

1. CORE	3
1.1 Release Notes	3
1.2 Схема БД CORE	5
1.3 Развёртывание платформы	6
1.4 Back-end	6
1.4.1 Обработка ошибок	6
1.4.2 Сервис для Поля ввода Mask	8
1.4.3 Стандарт разработки	9
1.5 Front-end	22
1.5.1 Конструктор	22
1.5.1.1 Атрибут	23
1.5.1.2 Класс	25
1.5.1.3 Объект	28
1.5.1.4 Страница	33
1.5.2 Интерфейс и элементы управления	41
1.5.2.1 Авторизация	41
1.5.2.2 Глобальная панель	42
1.5.2.2.1 Основное меню, панель закладок (быстрого доступа)	43
1.5.2.2.2 Панель оповещения	44
1.5.2.2.3 Профиль пользователя	44
1.5.2.2.4 Информация о программе	45
1.5.2.3 Рабочая область	45
1.5.2.4 Элементы управления и их обозначение	45
1.5.2.5 Элементы ввода данных	52
1.5.2.6 Окна для ввода данных	54
1.5.2.7 Мастер ввода данных (песочные часы)	55
1.5.3 Компоненты конструктора	56
1.5.3.1 Гриды	57
1.5.3.1.1 Простой грид (Grid)	58
1.5.3.1.2 Грид типа "Дерево" (Grid Tree)	60
1.5.3.2 Кнопки (Button)	60
1.5.3.2.1 Собиратель кнопок (Button Collector)	63
1.5.3.3 Колонки	63
1.5.3.3.1 Колонка "Boolean" (Column Boolean)	64
1.5.3.3.2 Колонка "Дата" (Column Date)	65
1.5.3.3.3 Колонки-кнопки	65
1.5.3.3.4 Текстовые колонки	68
1.5.3.3.5 Числовые колонки	69
1.5.3.4 Окно (Window)	70
1.5.3.4.1 Автособираемое окно	70
1.5.3.4.2 Кастомное окно	73
1.5.3.5 Панели	75
1.5.3.5.1 Динамическая панель (Dynamic Panel)	75
1.5.3.5.2 Контейнер типа "Вок"	79
1.5.3.5.3 Контейнер типа «Панель» (Panel)	80
1.5.3.5.4 Панель детализации в гриде (Panel Detail)	80
1.5.3.5.5 Панель загрузки и отображения файлов (Panel File)	81
1.5.3.5.6 Панель с историческими данными (Panel History)	81
1.5.3.5.7 Панель со вкладками (Tab Panel)	84
1.5.3.5.8 Панель с фильтрами (Filter Panel)	84
1.5.3.6 Поля	84
1.5.3.6.1 Групповые поля	86
1.5.3.6.2 Поле "Выпадающая таблица" (Field Grid)	86
1.5.3.6.3 Поле "Выпадающий список" (Field Combobox)	86
1.5.3.6.4 Поле "Изображение" (Field Image)	87
1.5.3.6.5 Поле "Песочные часы" (Field Item Selector)	88
1.5.3.6.6 Поле "Флаг" (Field Checkbox)	88
1.5.3.6.7 Поля с датами	88
1.5.3.6.8 Пустое пространство (Empty Space)	90
1.5.3.6.9 Текстовые поля	90
1.5.3.7 Стока в меню "Переход на страницу" (Menu Redirect)	91
1.5.3.8 графики	91
1.5.4 Работа с компонентами конструктора	91
1.5.4.1 Redirect извне с подстановкой фильтров	92
1.5.4.2 Redirect на страницу	92
1.5.4.3 Reusable objects (Переиспользуемые объекты)	94

1.5.4.4 Глобальные переменные	95
1.5.4.4.1 Применение глобальных переменных	95
1.5.4.5 Загрузка/выгрузка файлов по кнопке	98
1.5.4.6 Как настроить объект "Песочные часы"	98
1.6 Шлюз	99
1.6.1 Описание работы шлюза	99
1.6.2 Плагины шлюза	102
1.6.3 Разработчику	108
1.7 Система модулей	108

CORE

Поиск документации

Избранные страницы

Содержимое по меткам

Содержимое с указанными метками не найдено.

Популярные разделы

- [file-list](#)
- [project-poster](#)

Недавно обновлённые страницы

- [Система модулей](#)
2 мин. назад • обновлен о Анастасия Романюк • изменения
- [Back-end](#)
5 мин. назад • обновлен о Анастасия Романюк • изменения
- [Разработчику](#)
5 мин. назад • обновлен о Анастасия Романюк • изменения
- [Шлюз](#)
6 мин. назад • обновлен о Анастасия Романюк • изменения
- [Применение глобальных переменных](#)
6 мин. назад • обновлен о Анастасия Романюк • изменения

[Показать больше](#) 

Release Notes

Core 2.0

Важное

1. *Импортозамещение наше всё.*
[CORE 2.0 теперь на Postres](#). Все сервисы теперь будут храниться в таблице s-mt.t_query БД Postgres.
2. [Внутренняя Авторизация](#).
Теперь можно создавать пользователей, роли и доступные действия прямо в CORE.
3. Все изменения внесли в схему БД CORE 2.0 ([тут](#))
4. Не знаете, как развернуть платформу?
А у нас и [Инструкция](#) есть ([тут](#))

Новые компоненты

1. Теперь можно рисовать графики.
Добавили классы [Chart Conteiner](#) и [Chart](#). Можно изменять тип графиков, их цвет, местоположение легенды.



2. Текст в поле должен быть обязательно **красненьким** и **курсивом**?

Встречайте расширенный текстовый редактор - Field WYSIWYG. Теперь можно разукрашивать текст разными цветами,

менять его стиль и размер, создавать нумерованные списки и таблицы. И даже смайлики можно



Сохранить ОТМЕНА Редактирование

WYSIWYG

1. даны
2. две)

3. Нужно разместить стихотворение или просто в тексте многобукв?

Используйте класс Field Textarea, который позволяет вводить и отображать многострочный текст.

Textarea

Ехал Грека через реку.
Видит Грека в реке рак.
Сунул в реку руку Грека.
Рак за руку Греку - цап.

4. Нужно вставить картинку с котиками?

Добавили класс Field Image.



5. Хотите подглядывать за другим сайтом?

"Окно в Европу" - класс IFrame для создания фреймов на странице.

Iframe

Материал из Википедии — свободной энциклопедии

iFrame, iframe:

- [iFrame](#) — формат цифрового видео, разработанный компанией Apple.
- <iframe> (от англ. *inline frame*) — один из двух тегов языка HTML, обозначающих **фрейм** (отдельный, законченный HTML-документ, который вместе с другими HTML-документами может быть отображён в окне **браузера**).

См. также [править | править код]

- [i-кадр \(англ. i-frame\)](#)

Список значений слова или словосочетания со ссылками на соответствующие статьи.

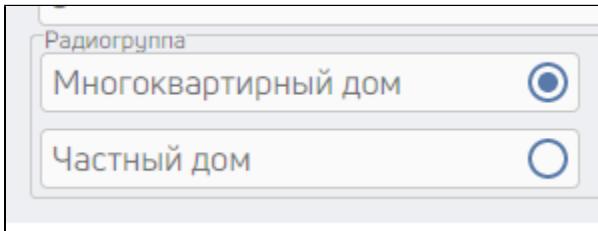
Если вы попали сюда из текста другой статьи Википедии, пожалуйста, вернитесь и уточните ссылку так, чтобы она указывала на нужную статью.

Категория: Страницы значений по алфавиту

Эта страница в последний раз была отредактирована 30 августа 2016 в 09:15.
Текст доступен по [лицензии Creative Commons Attribution-ShareAlike](#); в отдельных случаях могут действовать дополнительные условия. Подробнее см. [Условия использования](#).
Wikipedia® — зарегистрированный товарный знак некоммерческой организации [Wikimedia Foundation, Inc.](#).

6. А можно всех посмотреть?

Добавили класс Field Radio Group, который позволяет выбирать одно из нескольких представленных значений.



Новый функционал

1. Широкую на широкую!

Реализовали Drag-and-drop вкладок. Теперь можно изменять порядок открытых страниц, просто перетащив нужную вкладку.

2. Добавили настройки отображения колонок грида.

Теперь можно скрывать ненужные колонки. А если вдруг опять понадобятся - вновь отображать. НЕ нужна эта настройка? Скройте кнопку, указав `btnsettings = false` в атрибутах грида.

3. Реализовали окно редактирования значения behavior-атрибутов.

При заведении условия можно сразу провалидировать его выполнение.

4. Мультизагрузка файлов.

Если нужно загрузить сразу несколько файлов, не нужно грузить их по одному.

5. Динамические отчеты.

Позволяют заводить отчеты через интерфейс на основе SQL-запроса и без создания шаблона в JasperStudio.

6. Реализовали возможность перехода на страницу CORE с подстановкой фильтров.

Для получения ссылки необходимо подставить зашифрованный JSON с фильтрами в URL вида: `http://адрес core/redirect /зашифрованный JSON`. Шифрование производится с помощью `encodeURIComponent(btoa(unescape(encodeURIComponent(str))))`, где str - JSON с фильтрами.

7. Реализовали управление областью видимости на странице - Вертикальный сплиттер между объектами в горизонтальных панелях (класс Panel).

Для включения сплиттера необходимо задать значения атрибутов панели `resizeable=true` и `contentview=hbox`.

Минимальная ширина одного объекта в панели = 10% от ширины экрана.

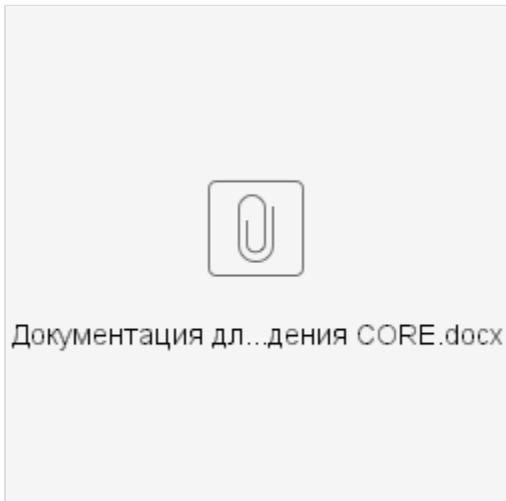
Схема БД CORE

Схема ErWin БД CORE



Развертывание платформы

Инструкция по развертыванию платформы CORE



Back-end

Эта страница описывает правила разработки и кодирования

- Обработка ошибок
- Сервис для Поля ввода Mask
- Стандарт разработки

Обработка ошибок

Эта страница описывает правила обработки ошибок на back-end и формирования ответа для передачи на front-end

- Формат возвращаемых данных
- Макросы / динамический текст
- Пакет pkg
- Пакеты вида pkg_json_*
- Пакеты вида pkg_*
- Пулы ошибок

Формат возвращаемых данных

Ошибка или список ошибок для возврата на фронт-энд должны быть оформлены в JSON таким образом, чтобы путь до них был \$.cv_error, внутри содержится массив значений

Примеры корректной выдачи:

```
"cv_error": {"23": []}
"cv_error": {"1001": [], "1003": [], "1306": []}
```

В скобках могут содержаться макросы динамических текстовок.

Макросы / динамический текст

Предусмотрено формирование ошибок, содержащих в себе макрос(ы) для подстановки кастомного текста.

Пример использования:

В s_mt.t_message:

```
ck_id = 14; cv_text = Существует объект "{0}", связанный через ИД мастер-объекта
```

Соответствующий вызов:

```
pkg.p_set_error(14, vv_texterror);
```

Т.е., текстовка самой ошибки может содержать макросы вида {0} .. {n} .. {4}

Метод pkg.p_set_error (а также p_set_warning и p_set_info) может принимать на вход до 5 макросов-текстовок включительно.

Пакет pkg

содержит в себе глобальные переменные, в которых хранятся списки ошибок, предупреждений и информационных сообщений - gv_error, gv_warning, gv_info. Также в этом же пакете находятся методы, используемые для обработки исключений, заполнения указанных глобальных переменных, их сброса, а также формирования итогового cv_error для передачи на фронт-энд.

Пакеты вида pkg_json_*

Функции таких пакетов должны содержать в себе:

1. В самом начале - вызов метода pkg.p_reset_response для очистки глобальных переменных
2. По ходу метода - логические проверки вида if pkg.gv_error is null then / if pkg.gv_error is not null then
3. При return для формирования cv_error - pkg.p_form_response (исключение для блока exception when others - можно оставить выдачу ошибки 512 в явном виде)

В итоге эти функции возвращают на фронт-энд строку, в которую склеивается текущие значения gv_error, gv_warning, gv_info

Пакеты вида pkg_*

Процедуры таких пакетов должны содержать в себе:

1. При реализации бизнес-логики - добавление ошибок через pkg.p_set_error(n), а также предупреждений и информационных сообщений через pkg.p_set_warning, pkg.p_set_info

2. По ходу метода - логические проверки вида if pkg.gv_error is null then / if pkg.gv_error is not null then
3. В блоке exception - единообразную конструкцию следующего вида:

```
exception
when others then
  pkg.p_handle_exception(sqlcode);
```

В итоге эти процедуры не возвращают строку-ответ, но заполняют глобальные переменные gv_error, gv_warning, gv_info

Пулы ошибок

Все коды ошибок ведутся в таблице t_message. Помимо ошибок, в этой таблице также хранятся предупреждения и информационные сообщения, разделение по полю cv_type.

Логически таблица t_message разделена на пулы:

Границы пула кодов	Назначение пула кодов
<= -100	Не занято; пока что не планируется к использованию
от -1 до -99	Рекомендуется: Ошибки миграции/интеграции
0	Без ошибок, «успешно»
1-499	Ошибки модуля Метамодель (логические ошибки при работе с Конструктором)
500 - 999	Системные ошибки (очевидно неверные вызовы, обработка ожидаемых ORA при многопользовательской работе и т.п.), обрабатываемые ошибки/сообщения шлюза
1000	Неизвестная ошибка
> 1000	Обычные ошибки проекта (неверный ввод пользователем, неверный формат, не заполнено обязательное поле и т.п.)

Сервис для Поля ввода Mask

В случае если нужны уникальные маски ввода для данных. Сервис указывается в глобальных настройках "smart_mask_query".

Сервис должен возвращать поля:

ck_id	уникальный идентификатор
cv_name	наименование
cv_series_mask	маска серии если есть
cv_number_mask	маска номера
cv_series_mask_desc	сообщение в случае не правильной серии
cv_number_mask_desc	сообщение в случае не правильного номера

Пример:

ck_id	7
cv_name	Военный билет
cv_series_mask	ББ
cv_number_mask	999999

cv_series_mask_desc	Формат ввода серии документа "ББ", где Б - Обязательная русская заглавная буква
cv_number_mask_desc	Формат ввода номера документа "999999", где 9 - обязательная цифра

Стандарт разработки

Эта страница описывает правила разработки и кодирования, принятые в проектах, которые используют Метамодель в качестве платформы.

- Вводная
- Легенда
 - Именование объектов SQL
 - Именование объектов PL/SQL
 - Именование сервисов
- Правила комментирования
 - SQL
 - PL/SQL
 - Отчеты в Jasper Reports
- Общие правила написания кода
- JSON
- Сортировка, фильтрация, пагинация
- Формат дат
- Обработка на backend oracle
 - float-полей (pkg_json_*)
 - строковых полей (pkg_json_*)
 - поля cd_en (pkg_*)
 - Принцип недоверия фронту
- DDL-образцы
 - Create Table oracle
 - Create Sequence
- Рекомендуемые конструкции
 - Выборка данных по максимальной дате
- Форматы данных
 - Формат массива
 - Формат boolean
 - Формат number
 - Остальные форматы данных

Вводная

Объектам БД, сервисам и запросам необходимо присваивать имена, отражающие их назначение. Наименования объектов должны содержать символы английского алфавита, а также могут содержать некоторые служебные символы (_#\$_). Использование транслитерации не допускается (необходимо использовать только слова английского языка и их сокращения). При именовании объектов необходимо проверять смысловую корректность названий, используя следующие средства:

- <https://www.lingvolve.com/en-us>
- <https://en.oxforddictionaries.com>
- <http://dictionary.cambridge.org>
- <https://www.google.com> (слово/словосочетание в кавычках «» для анализа используемости такого слова; анализ релевантности выдачи)

Легенда

□ – необязательные для заполнения элементы

<смысловая константа> – сокращение действий, выполняемых или включаемых в объект, или тип данных

<группа> – логический префикс, объединяющий объекты по типу/модулю/слово. Допустимая длина префикса составляет от 2 до 4 букв включительно.

В проекте существует документ с принятыми префиксами/сокращениями. Произвольные названия групп (незафиксированные в проектном документе) запрещены.

<наименование> – наименование объекта без префикса

Именование объектов SQL

В алфавитном порядке:

CONTEXT – ctx_<наименование>

CONSTRAINTS – cin_<смысловая константа>_<наименование таблицы>_[номер]

<смысловая константа>:

C – check

P – primary key (не может иметь номер в конце)

U – unique key

R – referential integrity (foreign key)

Наименование таблицы указывается без префикса t_

Пример:

Таблица t_example, первичный ключ – cin_p_example, внешний ключ – cin_r_example_1

DATABASE LINK – dl_<схема на которую ссылается с префиксами>

Пример: dl_gv

Наименование таблицы указывается без префикса s_

DIRECTORY – dir_<наименование>

FUNCTION – f_[<смысловая константа>]_<наименование>

<смысловая константа>: get – получить информацию

set – модификация данных

check – проверка

INDEX – cin_<смысловая константа>_<наименование таблицы>_[номер]

<смысловая константа>: P – primary key (не может иметь номер в конце)

U – unique key

R – referential integrity (foreign key)

Для неуникальных индексов по нескольким столбцам <смысловая константа> не указывается, например таблица t_example индекс по нескольким столбцам таблицы cin_example_1

JAVA CLASS – JC_<наименование>

JAVA RESOURCE – JR_<наименование>

JAVA SOURCE – JS_<наименование>

JOB – job_<наименование>

JSON – j_

LIBRARY – lb_<наименование>

MATERIALIZED VIEW – mv_[<группа>]_<наименование>

PACKAGE – pkg_[<группа>]_<наименование> в oracle, в postgres именуем так схемы где лежат функции и процедуры

PACKAGE BODY – точно также, как и PACKAGE

PROCEDURE – p_[<смысловая константа>]_<наименование>

Применяются такие же требования, как и для FUNCTION

PUBLIC SYNONYM – запрещено создание такого типа синонимов в связи с ограничениями по информационной безопасности.

QUEUE – q_<наименование>

SCHEMA – s_<наименование>

SCHEDULE – job_<наименование>

SEQUENCE – seq_<наименование таблицы>

PRIVATE SYNONYM – имя синонима должно совпадать с именем целевого объекта

TABLE – t_[<группа>]_<наименование>

Произвольные названия групп (незадокументированные в проектном документе) запрещены.

В наименованиях таблицы запрещено использовать множественное число (необходимо подобрать синоним или назвать сущность более абстрагировано).

Максимальная длина имени таблицы составляет 24 символа (вместе с префиксом T_, без префикса – 22 символа).

TEMPORARY TABLE – tt_<группа>_<наименование>

Применяются такие же требования, как и для TABLE

TRIGGER – tr_<смысловая константа>_<наименование таблицы>

<смысловая константа>: b – before

a – after

c – compound

i – insert

u – update

d – delete

s – statement

Пример: tr_bi_table_name

OBJECT TYPE – ot_<группа>_<наименование>

COLLECTION TYPE – ct_<группа>_<наименование>

VIEW - v_[<группа>]_<наименование>

Применяются такие же требования, как и для TABLE

XML SCHEMA - x_<группа>_<наименование>

Наименование столбцов в таблицах, представлениях, коллекциях, массивах и объектных типах:

<смысловая константа>_<наименование>

<смысловая константа>:

Char – запрещено к использованию в oracle

Long – запрещено к использованию в oracle

Varchar – запрещено к использованию в oracle

Varchar2 – v => cv_data

Date – d => cd_date

Number | BIGINT – n => cn_number

CLOB | TEXT – c => cc_lob

BLOB | BYTEA – b => cb_lob

XMLType – x => cx_schema

Timestamp – t => ct_zona

Number(1) | SMALLINT – замена Boolean – l => cl_boolean (такое поле обязательно д.б. not null, допускается хранить только 0 или 1)

INTERVAL – i => ci_day_to_second

Reference list => cr_list (мини-справочник, тип данных может быть number или varchar2, обязательно создавать constraint на поле с проверкой допустимых значений)

Пользовательский тип – u => cu_<наименование пользователя типа>

Первичный ключ – k, всегда во всех таблицах называется ck_id, обязан присутствовать во всех таблицах

Вторичный ключ – k, формат:

ck_<наименование родительской таблицы>

(ссылается на первичный ключ родительской таблицы)

Наименование родительской таблицы указывается без префикса t_

Ссылка таблицы на саму себя должна называться ck_parent

Префикс для служебных параметров: j

Например, jl_sort, jn_offset

Именование объектов PL/SQL

Локальные переменные – v<смысловая константа>_<наименование>

Глобальные переменные – g<смысловая константа>_<наименование>

Параметры методов – p<смысловая константа>_<наименование>

<смысловая константа>: в данном контексте типа данных столбца

Char – запрещено к использованию в oracle

Long – запрещено к использованию в oracle

Varchar – запрещено к использованию в oracle, postgres v, vv_data

Varchar2 –oracle v, vv_data

Date – d, vd_date

Number | BIGINT – n, vn_number

CLOB | TEXT – c, vc_lob

BLOB | BYTEA – b, vb_lob

JSON – j, vj_json

XMLEType – x, vx_schema

TIMESTAMP – t, gt_zone

BOOLEAN | SMALLINT – l, vl_flag, исключение делается для функций, вызываемых из sql, там параметр может быть NUMBER(1), и для наименования переменных, связанных с логическими именами столбцов таблицы

Reference list => vr_list (мини-справочник, тип данных может быть number или varchar2)

CURSOR – cur, vcur_sql, наименование курсора всегда cur_, а наименование переменной курсора в цикле vcur_

Переменные, передаваемые в курсор, всегда имеют префикс r[тип данных]_

Глобальные курсоры пакета начинаются так же cur_

Sys_ref_cursor – ref наименование самого курсора, vref_ наименование переменной

EXCEPTION – ex, vex_exception_name

TYPE trt_record IS RECORD – для записей ot_record, имя переменной vot_record;

TYPE tct_collection IS TABLE – для коллекций ct_collection, имя переменной vct_collection;

Первичный ключ – k, обязан называться в формате vk_<наименование таблицы>

Пользовательский тип – u, обязан называться vu_<наименование пользовательского типа с префиксом>

Именование сервисов

Имена запросов-сервисов должны содержать структурированные префикс и функциональное описание действия, выполняемого сервисом. При этом следует придерживаться CamelCase.

Примеры того, как именуются сервисы в БФЛ:

- с "D" начинаются все внутренние справочники - DContract, DDomesticSystem, DGenderType, etc
- с "jNSI" начинаются все справочники ЕС НСИ (или другие обращения к НСИ) - jNSIBIK, jNSIGetAddrArea, jNSIGetAddrByMO, jNSIOKTMO
- сервисы для отображения гридов обычно именуются как "префикс слоя" + "Show" + "логическая составляющая": MOShowBuildings, CAShowPersons, OrgShowOrganizations
- все имена сервисов движка Конструктора (Метамодели) начинаются либо с MT, либо с Get

Правила комментирования

SQL

Комментарии к таблицам и полям должны соответствовать именам в ER-модели, эти комментарии должны быть заполнены в обязательном порядке.

PL/SQL

Описание пакета в заголовке в формате

-- Purpose: назначение пакета, основные функции, выполняемые пакетом

Описание функций и процедур в формате:

/* Краткое смысловое описание, 1-2 предложения, на русском или английском

Параметры, требующие пояснения

*/

Комментарии для курсоров (явных и неявных) обязательны. Формат любой, главное – описать смысл.

Комментарии для сложных блоков обязательны, главное описать общий смысл.

Комментарии глобальных переменных пакета обязательны.

Формат комментариев может быть изменен в зависимости от программного обеспечения, которое будет собирать документацию.

Отчеты в Jasper Reports

Все скрипты, содержащиеся в датасетах шаблона Jasper, необходимо комментировать следующим образом:

```
/*< ><      ><      >*/  
  
/* 2_6          . 50.02.03  50.06.03 */  
  
  
/*< ><      ><      Header>*/  
  
/* 2_6          . 50.02.03  50.06.03  */  
  
Header.  
  
  
/*< ><      ><      ,      ,      >< >*/  
  
/* 2_6          . 50.06.03  */  
  
Dataset.      ,  Dataset_mo_improvement  Dataset_room_part_living,  
.
```

В табличном скрипте все поля необходимо комментировать, согласно наименованиям этих полей в ФС. Называть поля в скрипте также нужно, согласно общей конвенции БФЛ, например cv_name_service, по возможности не называйте поля безликими поле_1, поле_2.

Переменные в шаблоне также должны быть названы согласно конвенции БФЛ. Н/р vk_nsi_org_department или vk_period

Сами шаблоны обязательно должны содержать описание в полях свойств Description и Report Name.

Общие правила написания кода

- Запрещено использовать типы данных char, varchar (для oracle), long
- Для столбцов таблицы, для хранения числовых значений, всегда использовать тип NUMBER(n,m)
- Запрос или пакет, содержащий несколько запросов, должен быть написан в одном формате: либо ANSI синтаксис, либо ORACLE синтаксис. В случаях, когда приходиться менять синтаксис для улучшения плана выполнения запроса, необходимо в комментариях указать причину.
- Если система пишется с нуля, необходимо всегда использовать или ANSI синтаксис, или ORACLE синтаксис

- Наименования любых объектов или переменных должно состоять из символов a-z, A-Z, 0-9,_; любые другие символы запрещены
- Столбцы в запросе всегда должны содержать псевдоним (alias) таблицы, которой они принадлежат, даже если это запрос к одной таблице
- Запрещено использовать * в запросах вместо перечня столбцов
- Запрещено в теле представления или материализованного представления использовать другое представление, принадлежащее этой же схеме
- В операции insert всегда необходимо описывать список столбцов:

Insert into table (id1,id2,id3)

- Для логических столбцов используется только number(1) где 1 = true, 0 = false.
- Обязательно делать CHECK, описывающий допустимые значения для всех логических столбцов
- Запрещено создавать методы (function и procedure) вне пакетов
- Запрещено использовать триггеры
- Единый пакет констант с наименованием PKG
- Наименование констант не подчиняется правилам наименования глобальных переменных и должно быть как можно короче и в верхнем регистре. Например pkg.I pkg.U pkg.D
- В отдельных select-ах указание схемы запрещено
- Alias при написании запросов формируется по первым буквам из имени таблицы, т.е.: t_object => o , t_page_object => po, etc.

JSON

Важную роль для Конструктора (метамодели) играет единый стандарт отдачи и получения параметров.

При выдаче данных с бэкенда на фронт всегда передается один параметр, который называется JSON и содержит данные в формате JSON (см. в качестве примере любой сервис из t_query).

При получение данных с фронтенда на бэк передаются следующие параметры:

- page_object - Ид объекта с т.з. t_page_object
- session - сессия (в сервисах доступны сессионные данные, например, sess_ck_id и sess_session)
- json - содержит смысловую нагрузку, здесь стоит описать подробнее, см ниже

При передаче с фронта на бэк параметр json может содержать в себе три блока данных:

- service - системная информация, в частности, cv_action
- data - сами данные
- filter - данные для фильтрации, сортировки и пагинации - см. ниже

Сортировка, фильтрация, пагинация

- Любой запрос должен содержать order by:
 - если запрос используется в гриде - *order by &SORT* (на бэке не сортируем, сортировка придет с фронта, но можно дописать свою сортировку после фронтовой):

order by &SORT, cn_field1, cv_field2)

- если запрос используется в ПИДЕ - *order by cd_st desc*
- если запрос используется в выпадающем списке - *order by ck_id* (может быть изменено согласно требованиям спецификации)
- Запрос для грида должен содержать макросы &sort и &filter
 - Если помимо условий по &filter есть еще какие-то, то макрос следует обернуть в скобки:

```

oracle:
where cs.ck_con = json_value(:json, '$.master.ck_id' returning number)
and ( &FILTER )

postgresql:
where cs.ck_con = (:json::json#>>'{master.ck_id}')::bigint
and ( &FILTER )

```

У объекта грида, заведенного в Конструкторе, обязательно должен быть заполнен атрибут orderproperty. В противном случае грид не сможет быть отображен (при условии, что к гриду уже привязан сервис). Благодаря этому в GUI будет всегда явным образом отображено, по какому полю отсортированы данные (даже в случае сортировки по умолчанию)

- Для корректной работы грида при добавлении новой записи или редактировании существующей при использовании фильтров/пагинации необходимо в сервис грида добавить фильтр по ck_id основной таблицы:

```

oracle:
/*##filter.ck_id*/and t.ck_id = json_value(:json, '$.filter.ck_id'
returning number)/*filter.ck_id##*/

postgresql:
/*##filter.ck_id*/and t.ck_id = (:json::json#>>'{filter.ck_id}')::bigint
/*filter.ck_id##*/

```

Если грид содержит многокомпонентные данные, то необходимо в Конструкторе для него заполнить атрибут refreshallrecords для обновления всех строк грида.

- Существует два вида пагинации: с отображением кол-ва страница и без отображения (за это отвечает атрибут paginatortype)

Пример JSON, получаемый с фронт-енда для сортировки, фильтрации и пагинации:

```

json:{"filter":{"cv_short":" ", "cv_inn":"5634623656", "jl_sort":
[ { "property":"cd_st", "direction":"ASC" } ], "jl_filter": [ { "operator":"
eq", "value": "2017-10-25 00:00:00", "property": "cd_st" } ], "jn_offset":10,
"jn_fetch":5 } }

```

В коде SQL запроса для грида это обрабатывается так:

```

oracle:
...
where 1=1
/*##filter.cv_short*/and lower(org.cv_short) like '%' || lower
(json_value(:json, '$.filter.cv_short' returning varchar2)) || '%'
/*filter.cv_short##*/
/*##filter.cv_inn*/and org.cv_inn like '%' || json_value(:json, '$.
filter.cv_inn' returning varchar2) || '%'/*filter.cv_inn##*/
)t
where &FILTER
order by &SORT
offset &OFFSET rows
fetch next &FETCH rows only

postgresql:
...
where true
/*##filter.cv_short*/and lower(org.cv_short) like '%' || lower(:json:::
json#>>'{filter.cv_short}') || '%/*filter.cv_short##*/
/*##filter.cv_inn*/and org.cv_inn like '%' || :json::json#>>'{filter.
cv_inn}' || '%'/*filter.cv_inn##*/
)t
where &FILTER
order by &SORT
offset &OFFSET rows
fetch next &FETCH rows only

```

Стоит обратить внимание на макросы /*##filter.name*/ + /*filter.name##*/ (идут парой, открывающий и закрывающий макросы соответственно).

Эти макросы должны быть использованы во всех случаях, когда фильтр не является обязательным.

Цель - облегчить запрос к БД (иначе приходится делать больше кол-во OR-ов в блоке where).

Эти макросы обрабатываются средним слоем, и, если фильтр не указан, то содержимое макроса просто стирается из выполняемого запроса.

Пример формирования полей на бэк-енде для работы пагинации:

В SQL запросе, в блоке select:

```

/*
row_number() over(order by &SORT)as jn_rownum,
count(1) over() as jn_total_cnt,

```

В результирующем JSON получается:

```

"jn_rownum":1,"jn_total_cnt":9

```

Формат дат

в JSON используется формат даты ISO 8601, выглядит он следующим образом:

yyyy-mm-ddThh:mm:ss

например:

2008-09-15T15:53:00

Такой формат даты используется на фронт-енде; он же получается, когда мы преобразуем в БД дату в json_object.

Пример:

```
oracle:
select json_object(
    ...
    key 'cd_st' value t.cd_st,
    key 'cd_en' value t.cd_en,
    ...
) as json
from table t
postgresql:
select json_build_object(
    'cd_st', to_char(t.cd_st, 'yyyy-mm-dd"T"hh24:mi:ss'),
    'cd_en', to_char(t.cd_en, 'yyyy-mm-dd"T"hh24:mi:ss')
) as json
from table t
```

При написании методов модификации в пакетах pkg_json_* следует внутри блока from() указывать, что это строка и преобразовывать её в дату:

```
oracle:
select to_date(jt.cd_st, 'yyyy-mm-dd"T"hh24:mi:ss'),
       to_date(jt_cd_en, 'yyyy-mm-dd"T"hh24:mi:ss')
  into pot_org_role.cd_st,
        pot_org_role.cd_en
   from json_table(pc_json, '$' columns
                                cd_st  varchar2(100) path '$.
data.cd_st',
                                cd_en  varchar2(100) path '$.
data.cd_en') jt;

postgresql:
select to_date(jt.cd_st, 'yyyy-mm-dd"T"hh24:mi:ss'),
       to_date(jt_cd_en, 'yyyy-mm-dd"T"hh24:mi:ss')
  into pot_org_role.cd_st,
        pot_org_role.cd_en
   from json_to_record(pc_json) as x(cd_st text, cd_en text);
```

При работе (получении даты из формата JSON) с датами в JSON Oracle предусматривает несколько вариантов:

1. Получение значения даты используя неявное преобразование, указав возвращаемый тип DATE или TIMESTAMP;
2. Получение даты в виде строки и преобразование её в формат даты используя функцию TO_DATE.

Второй вариант является более предпочтительным т.к. при конвертировании вы контролируете заданный формат и меньше шансов получить ошибку, чем при неявном преобразовании.

Так же в Oracle есть ошибка (или фича, пока тут не до конца понятно), когда делаешь получения значения даты используя неявное преобразование получается дата без времени, чтобы получить время необходимо указывать возвращаемый тип timestamp, что, наверное, не совсем корректно.

Примеры работы:

```
with
  w_json as
  (
    select '{datetime: "2008-09-15T15:53:00"}' f_json
      from dual
  )
select json_value(f_json, '$.datetime' returning
date)           as dt,
          json_value(f_json, '$.datetime' returning timestamp)   as
ts,
          to_date(json_value(f_json, '$.datetime' returning varchar2
(100)), 'yyyy-mm-dd"T"hh24:mi:ss') as dt_2
  from w_json;
```

DT	TS	DT_2
15/09/2008	15/09/2008 15:53:00,000000	15/09/2008 15:53:00

В примере выше показана особенность работы с датами при получении её из формата JSON, поэтому рекомендуется проводить преобразование самостоятельно.

Обработка на backend oracle

float-поля (pkg_json_*)

В пакетах pkg_json_* по тем полям, которые подразумевают возможность ввода дробных чисел в GUI, следует применять следующую обработку:

```
-- JSON -> rowtype
select ...
  to_number(trim(jt.cn_loss_pct), rpad(lpad('D', 38, '9'), 63,
'9'), 'NLS_NUMERIC_CHARACTERS = ''.,''''),
  to_number(trim(jt.cn_loss_value), rpad(lpad('D', 38, '9'), 63, '9'),
'NLS_NUMERIC_CHARACTERS = ''.,''''),
  ...
into ...
  pot_point.cn_loss_pct,
  pot_point.cn_loss_value,
  ...
from json_table( ... )
```

Как правило, такими полями являются: проценты, тарифы, величины, потери и т.д.

строковых полей (pkg_json_*)

В пакетах pkg_json_* по тем полям, которые подразумевают возможность ввода нецифровых символов в GUI, следует явно указывать JSON_table_on_error_clause как "error on error":

<https://docs.oracle.com/database/121/SQLRF/functions092.htm#SQLRF56973>

```
-- JSON -> rowtype
select ...
  from json_table(pc_json,
                  cv_number varchar2(4000 char) path '$.data.
cv_number' error on error ,
...
...
```

Причина: по умолчанию применяется *null on error* и, если с фронт-енда приходит строка, длина которой больше допустимой, то в БД попадет null, никакой ошибки не произойдет. При *error on error* будет выдано сообщение об ошибке.

поля cd_en (pkg_*)

В пакетах pkg_*, где происходит обработка поля cd_en, сохранение данных в это поле всегда должно происходить со временем 23:59:59. Следует применять следующую обработку:

```
...
if pot_mo_service_norm.cd_en is not null then
  pot_mo_service_norm.cd_en := trunc(pot_mo_service_norm.cd_en, 'DD')
+ to_dsinterval('0 23:59:59');
end if;
```

Принцип недоверия фронту

При обработке данных на бэке необходимо руководствоваться «принципом недоверия» фронту и шлюзу.

1. Обязательность полей, формат данных и т.д. – нужно всегда проверять на бэке
2. Если на фронте поле ввода активно только при заведении данных, а потом при редактировании заблокировано – надо не брать значение с фронта, а взять существующее из БД, т.к. данные с фронта могли быть скомпрометированы

DDL-образцы

Create Table oracle

При создании таблиц следует всегда использовать опцию ROWDEPENDENCIES (https://docs.oracle.com/cd/B19306_01/server.102/b14200/pseudocolumns007.htm)

Не следует описывать tablespace и т.п.

Пример:

```
create table schema.table_name
(
...columns...
) rowdependencies;
```

Create Sequence

Валидный пример:

```
create sequence schema.seq_name
minvalue 1
nomaxvalue
start with 1
increment by 1
nocache;
```

Рекомендуемые конструкции

Выборка данных по максимальной дате

Было:

```
--CASHowRoomsByMO OLD
select      mr.ck_id,
            mr.ck_mo,
            mr.cv_number
        from t_mo mo
        join t_mo_room mr on mr.ck_mo = mo.ck_id
                           and mr.cd_st = (select max(cd_st) from t_mo_room
where ck_mo = mo.ck_id)
        where mo.ck_parent = json_value(:json, '$.master.ck_id' returning
number)
        order by mr.cv_number asc
```

Рекомендуется делать следующим образом:

```
--CASHowRoomsByMO
Select ck_id,
       ck_mo,
       cv_number
    from (
select mr.ck_id,
       row_number() over (partition by mr.ck_mo order by(mr.cd_st)
desc ) as cn_rn,
       mr.ck_mo,
       mr.cv_number
    from t_mo_room mr
   join t_mo mo
     on mr.ck_mo  = mo.ck_id
```

```
    where mo.ck_parent = json_value(:json, '$.master.ck_id' returning
number))
    where cn_rn = 1
```

Причина: убирается второй проход по t_mo_room

Форматы данных

Формат данных распространяется при передаче с фронт части на сервер. При передаче данных с сервера желательно следовать этим же правилам, за исключением, что пустые значения могут не передаваться, т.е. принимать значения undefined

Формат массива

1. При передачи пустого массива [] или пустоты (undefined) необходимо преобразовывать в (null)

Формат boolean

1. При незаполненном поле необходимо преобразовывать в (0) числовой.
2. Данные могут передаваться только в 2х логических вариантах (0) или (1)

Формат number

1. Может быть передано в 2х форматах: string или number. String формат передается, при условии что значение пустое или превышает допустимое количество значений (BigNumber)

Остальные форматы данных

1. При незаполненном поле (undefined, null, "") необходимо преобразовывать в пустое значение ("")

Front-end

Каталог для описания компонентов и настроек Front-end

Конструктор

Конструктор – фреймворк, т.е программная платформа, определяющая структуру программного проекта; облегчает разработку и объединение нескольких компонентов проекта.

- Атрибут
- Класс
- Объект
- Страница

Цель создания конструктора – упростить и ускорить разработку программного продукта. Конструктор предоставляет иерархию и набор типовых компонентов, из которых можно "собрать" любой программный продукт.

Структура метамодели

Основные компоненты:

1. Класс;
2. Объект;
3. Страница.



Атрибут

Атрибут – свойство класса либо объекта.

Каждому классу соответствует набор атрибутов.

- Создание атрибута
- Типы атрибутов
- Редактирование атрибута
- Удаление атрибута

Создание атрибута

Для создания атрибута необходимо:

1. перейти на страницу «Атрибуты» (Меню -> CORE -> Конструктор GUI -> Классы);
2. нажать кнопку «Создать атрибут»;

Наименование	Описание	Тип атрибута
align	Расположение текста...	basic
allownew	Признак доступности ввода нового значения (по умолчанию = false)	basic
autoload	Признак автозагрузки сервиса	basic
autoselectidentity	Имя атрибута, который отвечает за уникальность возвращаемых данных. Если этот параметр есть ...	basic
bottomleft	Служебный параметр для иерархии	placement
bitaudit	Показывать иконку с аудитом для грида	basic
breakOnEnter	Разрыв строки при нажатии клавиши Enter	basic

3. заполнить поля;
4. нажать кнопку «Сохранить».

Наименование	Описание	Тип атрибута
Наименование *	Описание *	Тип атрибута *
align	Расположение текста...	basic
allownew	Признак доступности ввода нового значения (по умолчанию = false)	basic
autoload	Признак автозагрузки сервиса	basic

Доступные поля для заполнения:

Наименование поля	Тип данных	Комментарии
Наименование	Текст	Обязательно для заполнения.

Описание	Текст
Тип атрибута	Выбор из списка

Типы атрибутов

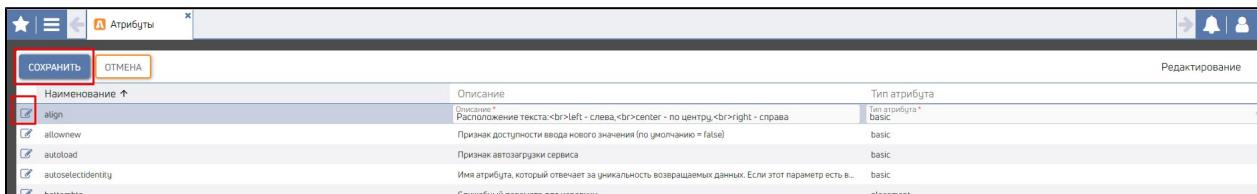
Выделяют несколько типов атрибутов:

Наименование	Описание
basic	обычный атрибут
behavior	поведенческий атрибут для зависимости отображения от данных. Переопределяется только на страницах.
system	системный атрибут, не переопределяется для объекта/страницы
placement	место для размещения дочернего элемента

Редактирование атрибута

Для редактирования атрибута необходимо:

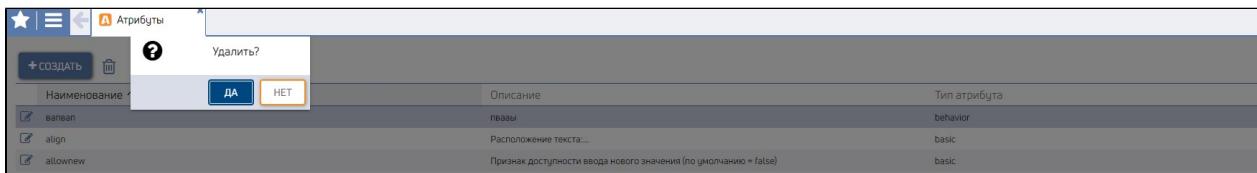
1. выбрать атрибут;
2. нажать на кнопку-колонку «Редактировать значение»;
3. внести необходимые изменения;
4. нажать кнопку «Сохранить».



Удаление атрибута

Для удаления атрибута необходимо:

1. выбрать атрибут;
2. нажать на кнопку «Удалить»;
3. подтвердить удаление, нажав на кнопку «Да».



Важно!

Если атрибут назначен какому-либо классу, то система вернет ошибку: «Атрибут используется в [Наименование класса]».

Для удаления такого атрибута необходимо удалить привязку атрибута классу, а затем выполнить удаление самого атрибута.

Класс

Класс – базовый компонент Конструктора.

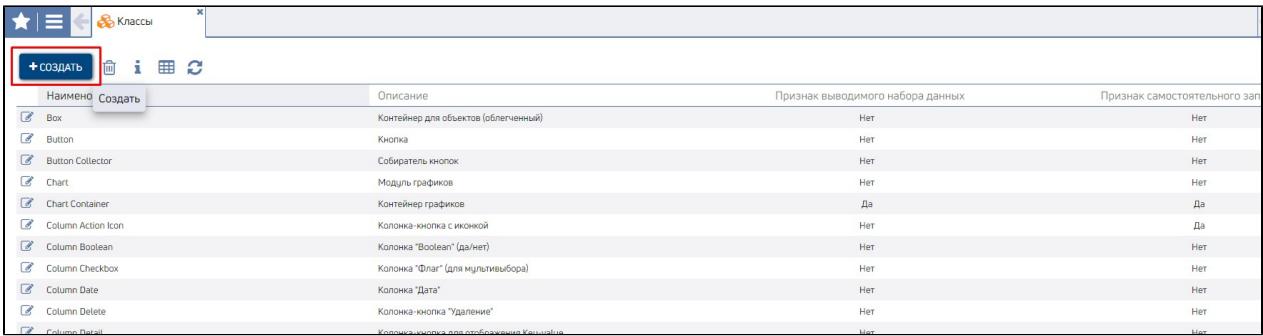
Представляет собой шаблон объекта с набором атрибутов и их значений.

- Создание класса
- Редактирование класса
- Удаление класса
- Добавление атрибутов класса
- Дочерние и родительские классы

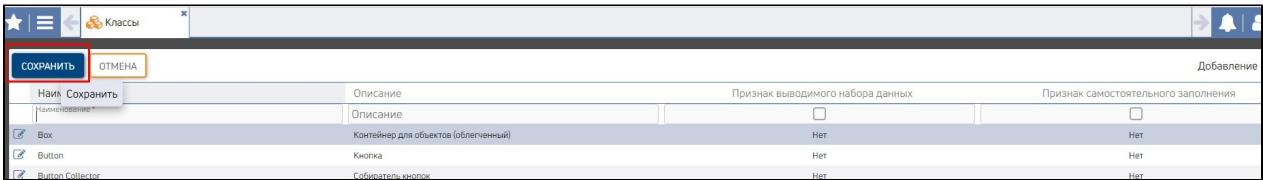
Создание класса

Для создания класса необходимо:

1. перейти на страницу «Классы» (Меню -> CORE -> Конструктор GUI -> Классы);
2. нажать кнопку «Создать»;



3. заполнить поля;
4. нажать кнопку «Сохранить».



Доступные поля для заполнения:

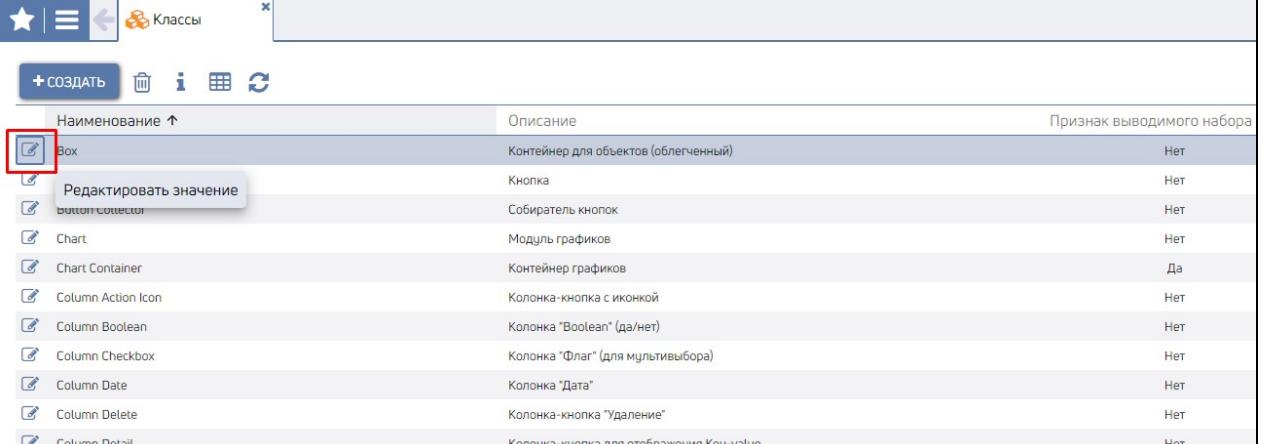
Наименование поля	Тип данных	Комментарии
Наименование	Текст	Обязательно для заполнения.
Описание	Текст	
Признак выводимого набора данных	Да/Нет	Можно ли привязать сервис к объекту класса.
Признак самостоятельного заполнения	Да/Нет	Может ли объект класса быть объектом верхнего уровня либо обязательно должен иметь родителя.

Редактирование класса

Для редактирования класса необходимо:

1. выбрать класс;

- нажать кнопку-колонку «Редактировать значение»;
- внести изменения;
- нажать кнопку «Сохранить».

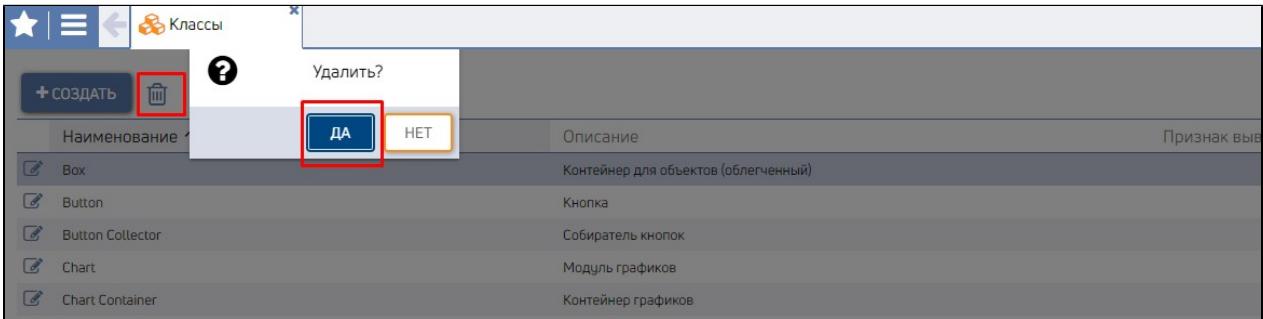


Наименование ↑	Описание	Признак выводимого набора
Box	Контейнер для объектов (облегченный)	Нет
Редактировать значение	Кнопка	Нет
Button Collector	Собиратель кнопок	Нет
Chart	Модуль графиков	Нет
Chart Container	Контейнер графиков	Да
Column Action Icon	Колонка-кнопка с иконкой	Нет
Column Boolean	Колонка "Boolean" (да/нет)	Нет
Column Checkbox	Колонка "Флаг" (для мультивыбора)	Нет
Column Date	Колонка "Дата"	Нет
Column Delete	Колонка-кнопка "Удаление"	Нет
Column Detail	Колонка-кнопка для отображения Key-value	Нет

Удаление класса

Для удаления класса необходимо:

- выбрать класс;
- нажать кнопку «Удалить»;
- подтвердить удаление, нажав на кнопку «Да».



Важно!

Если существует объект удаляемого класса, то система вернет ошибку: «Удаление класса невозможно, так как существуют связанные записи».

Для удаления такого класса необходимо удалить объект данного класса, а затем выполнить удаление самого класса.

Добавление атрибутов класса

После создания нового класса для него добавляют набор атрибутов. Для этого необходимо:

- выделить класс, для которого будут создаваться атрибуты;
- выбрать вкладку «Атрибуты»;
- нажать кнопку «Добавить»;

Код	Название	Нет	Нет
Button Collector	Собиратель кнопок	Нет	Нет
Chart	Модуль графиков	Нет	Нет
Chart Container	Контейнер графиков	Да	Да
Column Action Icon	Колонка-кнопка с иконкой	Нет	Да
Column Boolean	Колонка "Boolean" (да/нет)	Нет	Нет
Column Checkbox	Колонка "Флаг" (для мультивыбора)	Нет	Нет
Column Date	Колонка "Дата"	Нет	Нет
Column Delete	Колонка-кнопка "Удаление"	Нет	Нет
Column Detail	Колонка-кнопка для отображения Key-value	Нет	Нет
Column Edit	Колонка-кнопка "Редактирование"	Нет	Нет
Column Integer	Колонка "Целое число"	Нет	Нет

Атрибуты		Дочерние классы	Родительские классы
+ Добавить			
Наимен.	Добавить	Значение	Признак обязательности

Наимен.	Добавить	Значение	Признак обязательности	Описание	Тип атрибута
autoload		true	Да	Признак автозагрузки сервиса	basic
childs			Да	Служебный параметр для иерархии	placement
displayfield		cv_name	Да	Имя параметра, который будет отображен	basic
getglobalstore			Нет	Список глобальных переменных(через запятую), используемых в сервисе	behavior

4. заполнить поля;
 5. нажать кнопку «Сохранить».

Сохранение		Код класса		
Атрибуты		Дочерние классы	Родительские классы	
СОХРАНИТЬ		ОТМЕНА	Добавление	
Наименование	Сохранить	Значение	Признак обязательности	Описание
Наименование			<input type="checkbox"/>	Тип атрибута
autoload		true	Да	Признак автозагрузки сервиса
childs			Да	Служебный параметр для иерархии
displayfield			Да	Имя параметра, который будет отображен

Доступные поля для заполнения:

Наименование поля	Тип данных	Комментарии
Наименование	Выбор из списка	
Значение	Текст	
Признак обязательности	Да/Нет	
Описание		Заполняется автоматически при сохранении.
Тип атрибута		Заполняется автоматически при сохранении.

Дочерние и родительские классы

Иерархию классов можно настроить, добавив дочерние классы.

Для каждого класса можно привязать дочерние классы. Для этого необходимо:

- выделить класс, к которому будут добавляться дочерние классы;
- выбрать вкладку «Дочерние классы»;

3. нажать кнопку «Добавить»;

Наименование	Описание
Column Boolean	Колонка "Boolean" (да/нет)
Column Checkbox	Колонка "Флаг" (для мультивыбора)
Column Date	Колонка "Дата"
Column Delete	Колонка-кнопка "удаление"
Column Detail	Колонка-кнопка для отображения Key-value
Column Edit	Колонка-кнопка "Редактирование"
Column Integer	Колонка "Целое число"

4. заполнить поля;

5. нажать кнопку «Сохранить».

Доступные поля для заполнения:

Наименование поля	Тип данных	Комментарии
Атрибут класса	Выбор из списка	Перечень доступных значений зависит от выбранного родительского класса
Класс	Выбор из списка	Перечень доступных значений зависит от выбранного родительского класса

Список родительских классов выбранного класса отображается на вкладке «Родительские классы».

Наименование	Описание
Tab Panel	Панель со вкладками
Window	Окно
Wysiwyg Module	Модуль Wysiwyg

Объект

Объект – экземпляр класса с заданным набором атрибутов.

- Создание объекта
- Редактирование объекта
- Клонирование объекта
- Удаление объекта
- Переопределение значения атрибутов объекта

Создание объекта

Для создания объекта необходимо:

1. перейти на страницу «Объекты» (Меню -> CORE -> Конструктор GUI -> Объекты);
2. выбрать родительский объект либо корневой каталог и нажать кнопку «Создать»;

3. заполнить поля появившегося модального окна;
4. нажать кнопку «Сохранить».

Добавление Объекта

Класс *
Сервис
Имя объекта *
Описание объекта *
Порядок сортировки *
Отображаемое имя
Метод модификации
Провайдер данных при модификации

СОХРАНИТЬ ОТМЕНА

Доступные поля для заполнения:

Наименование поля	Тип данных	Комментарии
Класс	Выбор из списка	Обязательно для заполнения.
Сервис	Выбор из списка	
Имя объекта	Текст	Обязательно для заполнения.
Описание объекта	Текст	Обязательно для заполнения.
Порядок сортировки	Число	Обязательно для заполнения.
Отображаемое имя	Текст	
Метод модификации	Текст	
Провайдер данных при модификации	Выбор из списка	Заполняется только при указании Метода модификации.

Редактирование объекта

Для редактирования объекта необходимо:

1. выбрать объект;
2. нажать на кнопку-колонку «Редактировать значение»;

The screenshot shows a grid-based interface for managing objects. At the top, there are search and filter tools. Below is a header row with columns: Наименование объекта (Object Name), Описание (Description), Отображаемое имя (Visible Name), Имя сервиса (Service Name), Класс (Class), and Используется на страницах (Used on pages). The main body of the grid contains several rows of data. One specific row is highlighted with a red box around its first column, which contains a small icon followed by a 'grid tree' icon. To the right of this icon is a button labeled 'Редактировать значение' (Edit value), which is also highlighted with a red box.

3. внести необходимые изменения;
4. нажать кнопку «Сохранить».

The screenshot shows the 'Редактирование Объекта' (Edit Object) dialog box. It has a title bar and several input fields:

- Класс: Grid Tree (Грид-дерево для отображения данных)
- Сервис: MTObject
- Имя объекта*: SYS Grid Object << DO NOT CHANGE
- Описание объекта*: Объекты метамодели
- Порядок сортировки*: 1000400
- Отображаемое имя
- Метод модификации: pkg_json_meta.f_modify_object
- Провайдер данных при модификации*: Метамодель

At the bottom right are two buttons: 'СОХРАНИТЬ' (SAVE) in blue and 'ОТМЕНА' (CANCEL) in orange.

Клонирование объекта

Клонирование позволяет создать точную копию объекта со всеми его дочерними объектами и переопределенными значениями атрибутов. Наименование клонированного объекта будет иметь префикс "COPY of ".

Для клонирования объекта необходимо:

1. выбрать объект;
2. нажать на кнопку «Клонировать объект»;

The screenshot shows a list of objects in a catalog. One object, 'Test Grid', is selected and highlighted with a red box. A tooltip 'Клонировать объект' (Clone object) is displayed below the selection.

3. выбрать в появившемся модальном окне родительский объект, в котором будет создана клонированная копия объекта.

Если необходимо создать копию в корневом каталоге, то необходимо очистить поле «Родительский объект»;

The screenshot shows a 'Clone object' dialog box. The 'Parent object' field contains a search icon. The 'Parent object' dropdown menu lists several objects, with 'Account Grid' selected.

4. нажать кнопку «Клонировать».

The screenshot shows a 'Clone object' dialog box. The 'Clone' button is highlighted with a red box. An orange 'Cancel' button is also visible.

Важно!

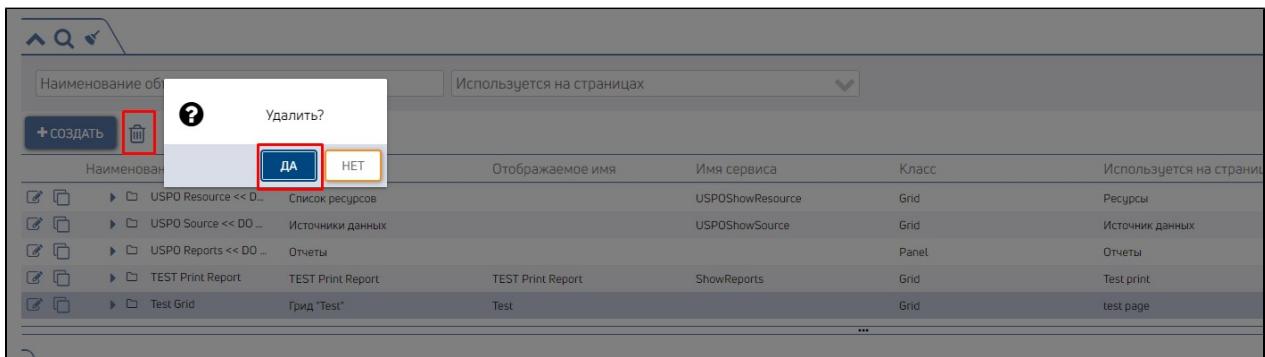
При клонировании объекта выполняется проверка на соответствие Иерархии классов.

Если класс клонируемого объекта не является дочерним для класса родительского объекта, система вернет ошибку: «Можно добавлять дочерние объекты только при наличии связи в т. С_Иерархия классов».

Удаление объекта

Для удаления объекта необходимо:

1. выбрать объект;
2. нажать на кнопку «Удалить»;
3. подтвердить удаление, нажав на кнопку «Да».



Важно!

Если объект используется на какой-либо странице, то система вернет ошибку: «Удаление Объекта невозможно, так как существует привязка объекта к странице».

Для удаления такого объекта необходимо [удалить привязку объекта на странице](#), а затем выполнить удаление самого объекта.

Переопределение значения атрибутов объекта

После создания объекта при необходимости переопределяются значения его атрибутов. Переопределенное значение атрибута на объекте приоритетнее значения атрибута на классе.

Для переопределения значения атрибута на объекте необходимо:

1. выбрать объект на странице «Объекты»;
2. выбрать нужный атрибут на вкладке «Атрибуты объекта» и нажать на кнопку-колонку «Редактировать значение»;

<input type="checkbox"/>	<input type="checkbox"/>	Tab Panel << DO ...	Таб-панель Страницы		Tab Panel	Страницы	30
<input type="checkbox"/>	<input type="checkbox"/>	Menu Redirect Column...	Menu Redirect Column Icon << Re...		Column Action Icon	Страницы	101
<input type="checkbox"/>	<input type="checkbox"/>	Scenario Grid	Сценарии автотестов	Сценарии	ATShowScenario	Сценарии	771
<input type="checkbox"/>	<input type="checkbox"/>	Scenario Step Grid	Шаги сценария	Шаги сценария	ATShowStep	Сценарии	772
<input type="checkbox"/>	<input type="checkbox"/>	Scenario Action Grid	Действия шага	Действия шага	ATShowAction	Сценарии	773
<input type="checkbox"/>	<input type="checkbox"/>	SYS Grid Page << DO ...	Страницы	MTPage	Grid Tree	Страницы	pkg_json_scenario_f_modify_page
<input type="checkbox"/>	<input type="checkbox"/>	SYS Grid Page Object ...	Объекты на странице	MTPageObject	Grid Tree	Страницы	1000 100
<input type="checkbox"/>	<input type="checkbox"/>						pkg_json_meta_f_modify_page_obj...
<input type="checkbox"/>	<input type="checkbox"/>						1000 300
					...		

Атрибуты объекта

Найменование	Значение по умолчанию	Значение (переопределено)	Описание
<input checked="" type="checkbox"/> autoload	true	false	Признак автозагрузки сервиса
<input checked="" type="checkbox"/> Редактировать значение	ok_id		Имя атрибута, который отвечает за уникальность возвращаемых данных...
<input checked="" type="checkbox"/> bnaudit	true		Показывать иконку с аудитом для грида
<input checked="" type="checkbox"/> bndelete	true		Показывать иконку с удалением для грида
<input checked="" type="checkbox"/> bthexcel	false		Показывать иконку с загрузкой excel для грида
<input checked="" type="checkbox"/> btrerefresh	true		Показывать иконку с обновлением для грида
<input checked="" type="checkbox"/> checkchildmore	true		Чекбокс для побывления еще одной записи

3. ввести переопределенное значение атрибута объекта;

4. нажать кнопку «Сохранить».

<input type="checkbox"/>	<input type="checkbox"/>	Scenario Step Grid	Сценарии шагов	ATShowStep	Grid	Сценарии	pkg_json_scenario_f_modify_step
<input type="checkbox"/>	<input type="checkbox"/>	Scenario Action Grid	Действия шага	ATShowAction	Grid	Сценарии	pkg_json_scenario_f_modify_action
<input type="checkbox"/>	<input type="checkbox"/>	SYS Grid Page << DO ...	Страницы	MTPage	Grid Tree	Страницы	pkg_json_meta_f_modify_page
<input type="checkbox"/>	<input type="checkbox"/>	SYS Grid Page Object ...	Объекты на странице	MTPageObject	Grid Tree	Страницы	pkg_json_meta_f_modify_page_obj...
<input type="checkbox"/>	<input type="checkbox"/>	SYS Grid Object << DO...	Объекты метамодели	MTOobject	Grid Tree	Объекты	pkg_json_meta_f_modify_object
<input type="checkbox"/>	<input type="checkbox"/>	SYS Grid Object Attrib...	Грид "Атрибуты объекта"	MTOobjectAttribute	Grid	Объекты	pkg_json_meta_f_modify_object_a...
<input type="checkbox"/>	<input type="checkbox"/>	SYS Grid Class << DO...	Таблица с классами	MTClass	Grid	ввввв. Классы	pkg_json_meta_f_modify_class
<input type="checkbox"/>	<input type="checkbox"/>	SYS Tab Panel Class ...	Панель со вкладками	Tab Panel	Классы	Классы	1000 900
<input type="checkbox"/>	<input type="checkbox"/>				...		

Атрибуты объекта

Найменование	Значение по умолчанию	Значение (переопределено)	Описание
<input checked="" type="checkbox"/> autoload	true	false	Признак автозагрузки сервиса
<input checked="" type="checkbox"/> autoselectidentity	ok_id		Имя атрибута, который отвечает за уникальность возвращаемых данных...
<input checked="" type="checkbox"/> bnaudit	true		Показывать иконку с аудитом для грида
<input checked="" type="checkbox"/> bndelete	true		Показывать иконку с удалением для грида

Для редактирования переопределенного значения атрибута объекта необходимо:

- выбрать атрибут;
- нажать на кнопку-колонку «Редактировать значение»;
- ввести новое значение;
- нажать кнопку «Сохранить».

Для удаления переопределенного значения атрибута объекта необходимо выбрать атрибут и нажать кнопку «Удалить».

<input type="checkbox"/>	<input type="checkbox"/>	Scenario Action Grid	Действия шага	ATShowAction	Grid	Сценарии	pkg_json_scenario_f_modify_action
<input type="checkbox"/>	<input type="checkbox"/>	SYS Grid Page << DO ...	Страницы	MTPage	Grid Tree	Страницы	pkg_json_meta_f_modify_page
<input type="checkbox"/>	<input type="checkbox"/>	SYS Grid Page Object ...	Объекты на странице	MTPageObject	Grid Tree	Страницы	pkg_json_meta_f_modify_page_obj...
<input type="checkbox"/>	<input type="checkbox"/>	SYS Grid Object << DO...	Объекты метамодели	MTOobject	Grid Tree	Объекты	pkg_json_meta_f_modify_object
<input type="checkbox"/>	<input type="checkbox"/>	SYS Grid Object Attrib...	Грид "Атрибуты объекта"	MTOobjectAttribute	Grid	Объекты	pkg_json_meta_f_modify_object_a...
<input type="checkbox"/>	<input type="checkbox"/>	SYS Grid Class << DO...	Таблица с классами	MTClass	Grid	ввввв. Классы	pkg_json_meta_f_modify_class
<input type="checkbox"/>	<input type="checkbox"/>	SYS Tab Panel Class ...	Панель со вкладками	Tab Panel	Классы	Классы	1000 900
<input type="checkbox"/>	<input type="checkbox"/>				...		

Атрибуты объекта

Найменование	Значение по умолчанию	Значение (переопределено)	Описание
<input checked="" type="checkbox"/> autoload	true	false	Признак автозагрузки сервиса
<input checked="" type="checkbox"/> autoselectidentity	ok_id		Имя атрибута, который отвечает за уникальность возвращаемых данных...
<input checked="" type="checkbox"/> bnaudit	true		Показывать иконку с аудитом для грида
<input checked="" type="checkbox"/> bndelete	true		Показывать иконку с удалением для грида

Страница

Страница - верхний компонент в иерархии объектов конструктора; включает в себя объекты и атрибуты.

Типы страниц:

- Модуль** - Абстрактное понятие, отражает крупную функциональную часть системы, включает в себя каталоги и страницы;

- Создание страницы
- Редактирование страниц
- Удаление страниц
- Сброс кэша
- Добавление объектов на страницу
- Редактирование объектов на странице

- 2. **Каталог** - смысловой раздел, который «группирует» страницы и вложенные каталоги;
 - 3. **Страница** - базовый раздел, содержащий объекты. Добавить объекты в каталог
- Удаление объекта со страницы
 - Мастера
 - Обновление привязки объектов на странице
 - Переопределение атрибутов объектов на странице
 - Глобальные переменные на странице

Создание страницы

Для создания страницы необходимо:

1. перейти на страницу «Страницы» (Меню -> CORE -> Конструктор GUI -> Страницы);
2. выбрать родительский модуль/каталог либо корневой каталог, в котором будет создана страница;
3. нажать кнопку «Создать»;

Название страницы	Код действия просмотра	Код действия модификации	Порядок
Страницы	511	512	1
Объекты	499	500	2
Классы	497	498	3
Атрибуты	491	492	4
Сервисы	515	516	5
Модули	515	516	10
Администрирование			2
Авторизация			3
...			

Объекты на странице

+ ДОБАВИТЬ | Удалить | Информация | Обновить

4. заполнить поля появившегося модального окна;

5. нажать кнопку «Сохранить».

Добавление Страницы

Наименование *
Тип *
Порядок *
Иконка
Код действия просмотра ?
Код действия модификации ?
Отображать в меню
Статическая страница
Ссылка
СОХРАНИТЬ
ОТМЕНА

Доступные поля для заполнения:

Наименование поля	Тип данных	Комментарии
Наименование	Текст	Обязательно для заполнения.
Тип	Выбор из списка	Обязательно для заполнения.
Порядок	Число	Обязательно для заполнения.
Иконка	Выбор из списка	
Код действия просмотра	Число	Заполняется только для типа «Страница».
Код действия модификации	Число	Заполняется только для типа «Страница».
Отображать в меню	Да/Нет	Отображать ли страницу (каталог/модуль) в Меню.
Статическая страница	Да/Нет	Является ли страница статической.
Ссылка	Текст	Адрес статической страницы. Заполняется только при значении поля «Статическая страница» = «Да». Доступные символы: [0-9],[a-z],[A-Z],[_],[-].

Создание каталога и модуля аналогичны созданию страницы.

Редактирование страниц

Для редактирования страницы необходимо:

1. выбрать страницу;

2. нажать на кнопку-колонку «Редактировать значение»;

The screenshot shows a list of pages in a catalog structure. The 'Редактировать значение' (Edit value) button is highlighted with a red box. The columns are 'Наименование страницы' (Page name), 'Код действия просмотра' (View action code), and a list of objects.

Наименование страницы	Код действия просмотра
Корневой каталог	
CORE	
Конструктор GUI	
Страницы	511
Редактировать значение	499
Классы	497
Атрибуты	491
Сервисы	515
Модули	515
Администрирование	
Авторизация	

3. внести необходимые изменения;
4. нажать кнопку «Сохранить».

Удаление страниц

Для удаления страницы необходимо:

1. выбрать страницу;
2. нажать на кнопку «Удалить»;
3. подтвердить удаление, нажав на кнопку «Да».

The screenshot shows a list of pages in a catalog structure. The 'Удалить' (Delete) button is highlighted with a red box. The columns are 'Наименование страницы' (Page name), 'Код действия просмотра' (View action code), and a list of objects.

Наименование страницы	Код действия просмотра
Корневой каталог	
CORE	
Меню	
Test	
Где я?	
цукава	
еееее	511
699	
test catalog	

Важно!

Если на страницу добавлен какой-либо объект, то система вернет ошибку: «Существуют объекты, привязанные к этой странице. Необходимо сначала удалить привязку»

Для удаления такой страницы необходимо удалить привязку объекта, а затем выполнить удаление самой страницы.

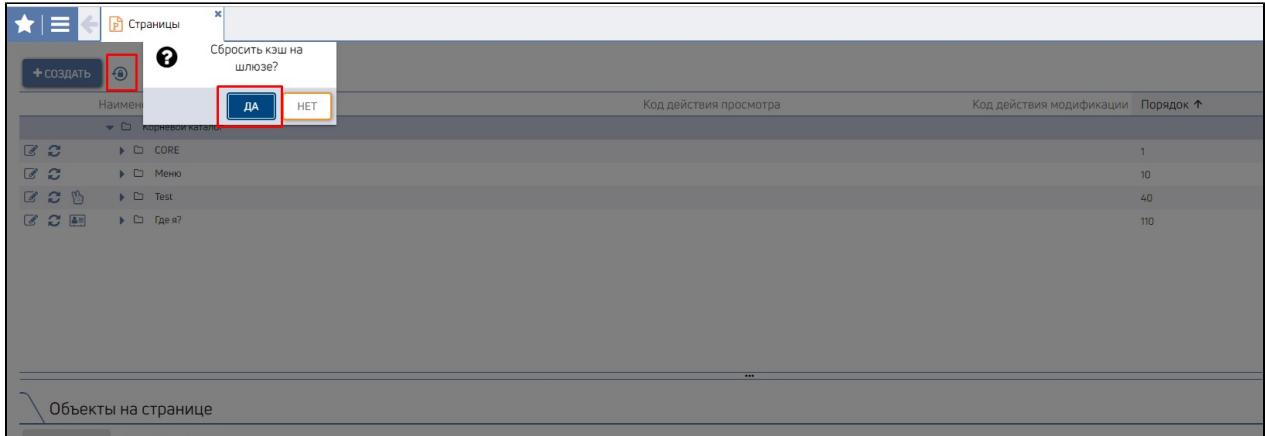
Сброс кэша

Для уменьшения количества запросов к БД и повышения быстродействия при первичном открытии страницы всё её наполнение кэшируется.

После изменений настроек страницы (редактирования самой страницы, добавлении/редактировании/удалении объектов на странице, обновления привязки и т.д.) необходимо осуществлять сброс кэша.

Для этого следует:

1. перейти на страницу «Страницы» (Меню -> CORE -> Конструктор GUI -> Страницы);
2. нажать на кнопку «Сбросить кэш на шлюзе»;
3. подтвердить сброс, нажав на кнопку «Да»;

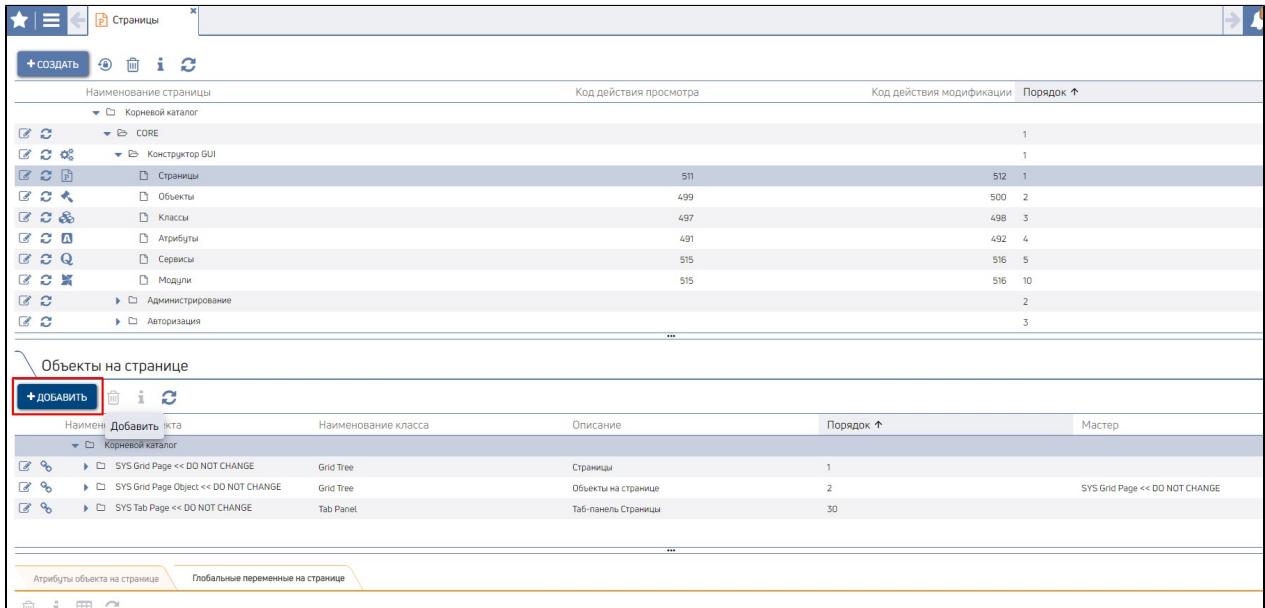


4. дождаться завершения технологических работ, связанных со сбросом.

Добавление объектов на страницу

После создания страницы можно приступить к добавлению объектов. Для этого необходимо:

1. выбрать созданную страницу;
2. нажать кнопку «Добавить»;



3. заполнить поля появившегося модального окна;

4. нажать кнопку «Сохранить».

Добавление Объекты на странице

Объект

Порядок*

Выбор мастера

СОХРАНИТЬ

ОТМЕНА

Доступные поля для заполнения:

Наименование поля	Тип данных	Комментарии
Объект	Выбор из списка	Обязательно для заполнения.
Порядок	Число	Обязательно для заполнения.
Выбор мастера	Выбор из списка	Используется для установки зависимостей между объектами.

Редактирование объектов на странице

Для редактирования объекта на странице необходимо:

1. выбрать объект;
2. нажать на кнопку-колонку «Редактировать значение»;
3. ввести новое значение;
4. нажать кнопку «Сохранить».

Удаление объекта со страницы

Для удаления объекта на странице необходимо:

1. выбрать объект;
2. нажать на кнопку «Удалить»;
3. подтвердить удаление, нажав на кнопку «Да».

Важно!

Если удаляемый объект либо его дочерний объект является мастером для другого объекта на странице, то система вернет ошибку: «Существует объект "наименование объекта", связанный через Ид мастер-объекта».

Для удаления такого объекта необходимо удалить связь с подчиненным объектом, а затем выполнить удаление самого объекта.

Мастера

В рамках одной страницы можно настроить зависимость объекта. Главный объект, влияющий на другие объекты, является мастером. Каждый объект может иметь только одного мастера. Мастер может иметь неограниченное количество зависимых объектов на странице.

Взаимодействие мастера и подчиненного объекта можно настроить с помощью атрибутов:

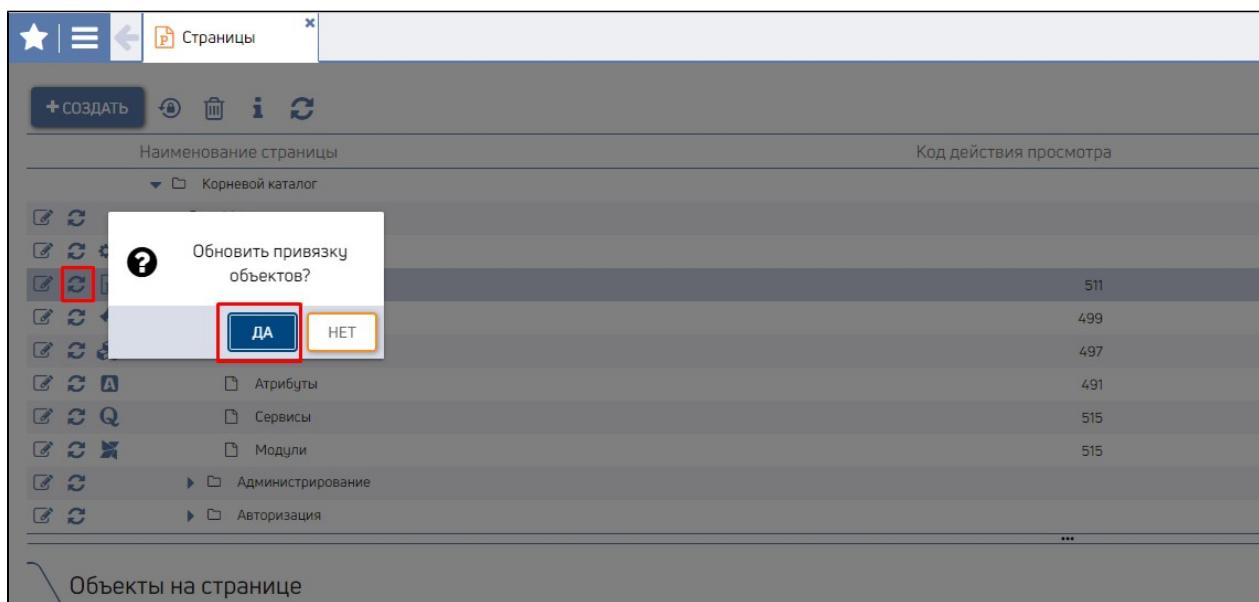
Наименование	Описание
disabledemptymaster	Флаг блокировки, если мастер пустой
idproperty	Наименование атрибута, по которому фильтруем значение от мастера
reloadmaster	Признак обновления родителя по мастеру
reloadservice	Параметр разрешающий повторное обновление комбобокса во время раскрытие в случае, если изменился мастер
reqsel	Признак зависимости от мастера (true/false)

Обновление привязки объектов на странице

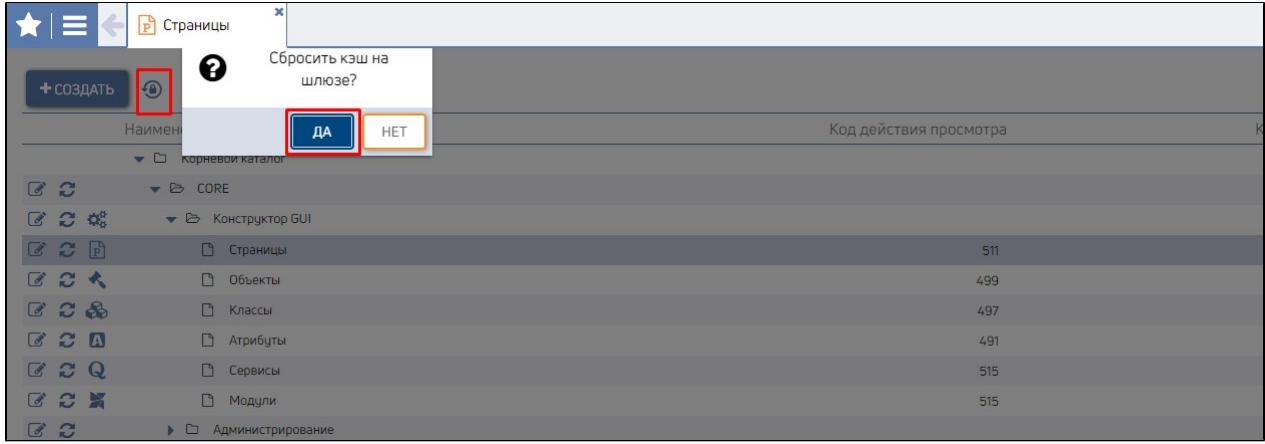
В случае, если в объект, который уже используется на странице, были внесены изменения (например, добавлены новые дочерние объекты) необходимо обновить привязку объектов на странице. В этом случае все внесенные изменения отобразятся на странице.

Для обновления привязки необходимо:

1. выбрать страницу, на которой необходимо обновить привязку объектов;
2. нажать на кнопку-колонку «Обновить привязку объектов»;
3. подтвердить обновление, нажав на кнопку «Да»;



4. нажать кнопку «Сброс кэша на шлюзе»;
5. подтвердить сброс, нажав на кнопку «Да».



Переопределение атрибутов объектов на странице

После добавления объектов на страницу и настройки мастеров можно, при необходимости, переопределить значения атрибутов объектов на странице. Переопределенное значение атрибута объекта на странице приоритетнее значения атрибута на объекте.

Переопределение значения атрибута объекта на странице аналогично переопределению значения атрибута на объекте ([ссылка](#)).

Глобальные переменные на странице

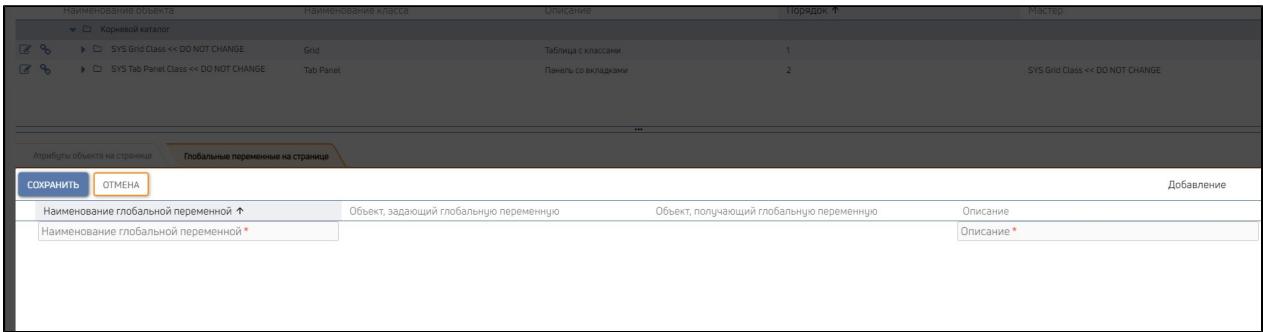
В рамках одной страницы необходимо задать список глобальных переменных. [Подробнее](#)

Для создания глобальной переменной на странице необходимо:

1. выбрать страницу;
2. перейти на вкладку «Глобальные переменные на странице»;
3. нажать кнопку «Добавить»;

4. заполнить поля;

5. нажать кнопку «Сохранить».



Для редактирования глобальной переменной необходимо:

1. выбрать глобальную переменную;
2. нажать на кнопку-колонку «Редактировать значение»;
3. ввести новое значение;
4. нажать кнопку «Сохранить».

Для удаления глобальной переменной необходимо выбрать глобальную переменную и нажать кнопку «Удалить».

Важно!

Если глобальная переменная используется на странице, то система вернет ошибку: «Удаление переменной невозможно, так как она все еще используется на странице».

Для удаления такой глобалки необходимо удалить привязку у объектов, задающих/получающих эту глобальную переменную, а затем выполнить удаление самой глобалки.

Интерфейс и элементы управления

Пользовательский интерфейс (графический интерфейс) Конструктора состоит из стандартных элементов, таких как: кнопки, меню, вкладки, фильтры, панели навигации и управления и т.п.

Пользовательский интерфейс выполнен в двух вариантах: «Светлая тема» и «Темная тема». Выбор темы не влияет на выполнение функций и сделан для удобства конкретного пользователя. Основные различия тем – цветовые, шрифтовые и компонентные решения, а также расположение панелей управления. По умолчанию, при первичной авторизации пользователя подгружается «Светлая тема», при последующих авторизациях – выбранная пользователем тема.

- Авторизация
- Глобальная панель
 - Основное меню, панель закладок (быстрого доступа)
 - Панель оповещения
 - Профиль пользователя
 - Информация о программе
- Рабочая область
- Элементы управления и их обозначение
- Элементы ввода данных
- Окна для ввода данных
- Мастер ввода данных (песочные часы)

Авторизация

Для входа в систему необходимо пройти авторизацию на стартовой странице системы. В поля ввода ввести логин и пароль, нажать кнопку

Технологическая платформа

Имя пользователя

Пароль

войти

«Войти». Пока поля не будут заполнены, данная кнопка будет неактивна.

Технологическая платформа

Имя пользователя

Пароль

войти

Технологическая платформа

admin_core

.....

войти

Символы введённого пароля будут скрыты.

Технологическая платформа

admin_core

.....

войти

Технологическая платформа

admin_core

123456

войти

Для возможности просмотра пароля необходимо навести курсор на поле ввода пароля, справа отобразятся значки «глаз» и «крестик». Нажать на значок «глаз». Скрыть пароль можно повторным нажатием на «глаз».

Технологическая платформа

admin_core

123456

войти

Глобальная панель

Конструктор имеет глобальную статичную панель, расположенную в верхней части страницы и состоит из следующих элементов (слева направо):

1. Кнопка вызова раздела быстрого доступа
2. Кнопка вызова раздела основного меню
3. Кнопки навигации по вкладкам
4. Область вкладок
5. Кнопка вызова раздела оповещений
6. Кнопка вызова профиля пользователя
7. Кнопка вызова информации о программе



1. Кнопка вызова раздела быстрого доступа
Кнопка осуществляет вызов бокового меню с вкладками, которые будут помечены пользователем как «Избранные». При отсутствии таких вкладок, боковое меню будет пустым. При первоначальной авторизации пользователя панель также будет пустой. Для того, чтобы в меню отображались «Избранные» вкладки, необходимо пометить их в основном меню.
2. Кнопка вызова раздела основного меню
Кнопка осуществляет вызов основного бокового меню со всеми вкладками программы. Здесь осуществляется переход ко всем необходимым разделам, для удобства все функции сгруппированы по категориям.
3. Кнопки навигации по вкладкам
Предназначены для перемещения по вкладкам, вправо и влево соответственно, если все активные вкладки не

помещаются в область вкладок, отображаемую браузером пользователя.

4. Область вкладок

Предназначена для быстрого переключения между активными разделами. Активная вкладка выделяется на панели другим цветом.

5. Кнопка вызова раздела оповещений

Кнопка осуществляет вызов информации по возникающим в процессе работы предупреждениям, ошибкам, системной информации. Данный раздел необходим для быстрой диагностики работы программы и упрощению поддержки пользователя при работе с программой.

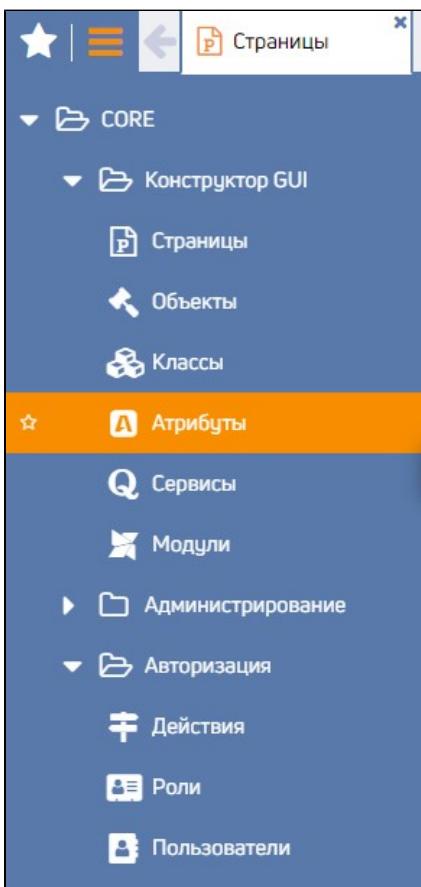
6. Кнопка вызова профиля пользователя

Кнопка открывает информацию о пользователе, осуществляющем работу с программой, а также позволяет осуществить выбор подразделения, периода и темы оформления.

7. Кнопка вызова информации о программе

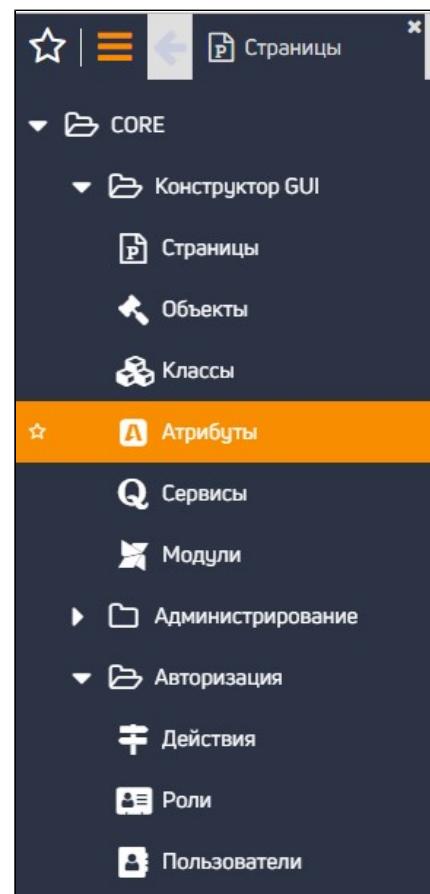
Кнопка вызывает информацию о версии программы и разработчике.

Основное меню, панель закладок (быстрого доступа)

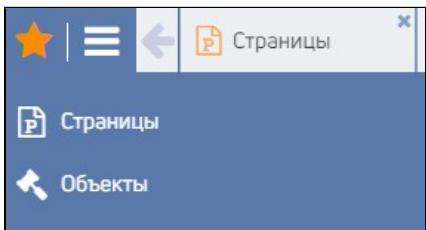
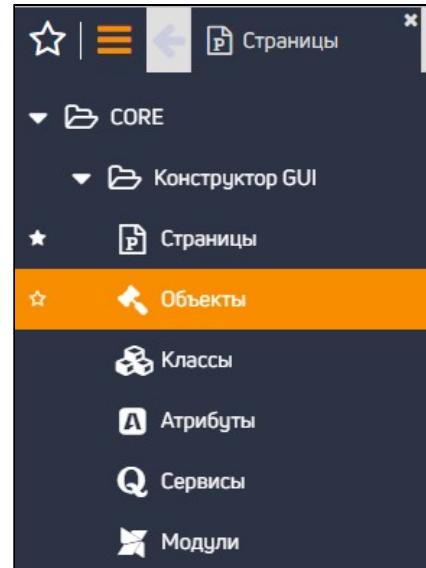
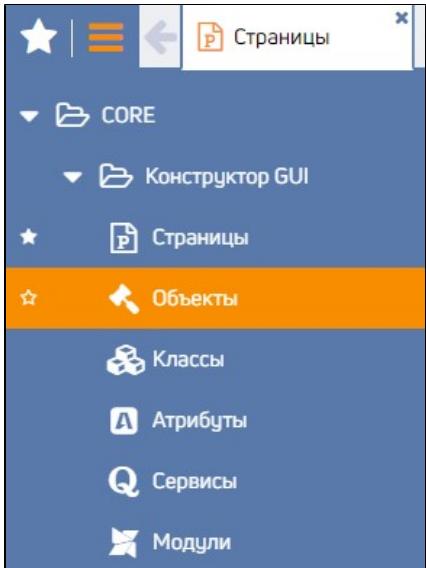


Основное меню визуально представляет собой дерево каталогов.

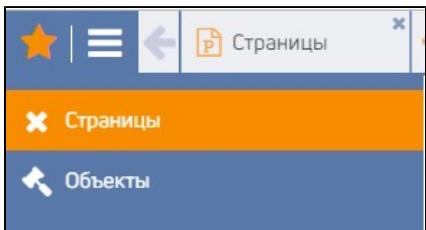
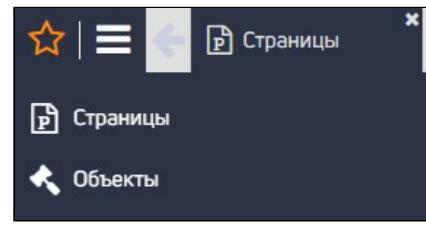
Каталог раскрывается при нажатии на название. Если в каталоге вложены ещё каталоги, то раскрытие будет происходить до достижения конечной страницы. Выбор страницы также происходит нажатием на название. Каталог имеет значок «Папка», при раскрытии каталога значок меняет свой вид на «раскрытый», а стрелка, расположенная слева от значка, меняет направление и указывает вниз. При скрытии содержимого каталога, значок меняет вид с «раскрытый» на «скрытый», стрелка указывает вправо.



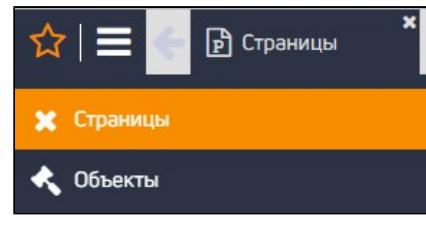
Для добавления страниц в панели быстрого доступа «Избранное» необходимо навести курсор на страницу, слева появится значок «звёздочка». Нажать на этот значок. «Звёздочка» выделится и будет постоянно отображаться напротив названия страницы в основном меню. Для снятия значка необходимо нажать на него повторно. При снятии значка, страница будет удалена из панели «Избранное»



При добавлении страниц в панель быстрого доступа «Избранное», страницы будут отображены не иерархией, а списком. Расположение позиций в списке будет изменяться по мере добавления позиций в зависимости от основной иерархии страниц по каталогам.



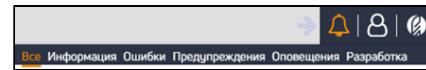
Для удаления страниц из панели быстрого доступа «Избранное» необходимо навести курсор мыши на графический значок страницы слева от названия, значок поменяет свой вид на «крестик», нажать его, страница будет удалена.



Панель оповещения



Для просмотра системной информации необходимо открыть меню «Оповещения».



Здесь содержится информация для разработчиков, системного администратора и технической поддержки. Здесь можно переключаться по вкладкам и типам оповещений.

Профиль пользователя

Для просмотра информации об авторизованном в системе пользователе необходимо открыть профиль пользователя. Также здесь можно сменить тему интерфейса.



Информация о программе

Информация о версии и сборке программы отображается во всплывающем окне после нажатия на кнопку «Информация о программе».

Рабочая область

Рабочая область представляет собой страницы, на которых расположены фильтры, связанные таблицы с вкладками, панели управления.

Представление рабочей области отличается в светлой и тёмной темах.

Связь между таблицами осуществляется по принципу «Аккордеон» - при выборе элемента в верхней таблице (блоке данных), отображается информация по выбранной записи в нижней таблице (блоке данных), если таблиц несколько, необходимо в каждой таблице (блоке данных) выбирать записи последовательно сверху вниз. В случае, если запись в верхней таблице выбрана не будет, панели управления на нижней таблице будут неактивны.

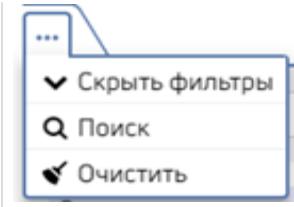
Элементы управления и их обозначение

Основные элементы управления представлены для каждого блока, с которым производятся действия и являются стандартными. Сгруппированы элементы также по блокам. Расположение в Тёмной теме – вертикальное, в Светлой теме – горизонтальное. В Светлой теме некоторые функциональные кнопки подписаны, в Тёмной теме используются интуитивные иконки, поэтому для уточнения назначения кнопки на ней следует навести курсор, появится всплывающая подсказка с наименованием кнопки.

Глобальные фильтры всегда расположены вверху рабочей области страницы.

Таблица 1 – Панель управления для блоков фильтрации

Активный элемент	Неактивный элемент	Описание элемента	Активный элемент	Неактивный элемент
		Собиратель кнопок (кнопка «еще»).	<div style="background-color: #0070C0; color: white; padding: 5px; text-align: center;"> Скрыть фильтры Поиск Очистить </div>	



Элемент предна значен для скрытия функциональных кнопок фильтра. Для отображения всех возможностей фильтра, требуется нажать на элемент, после чего раскроется список скрытых функций фильтрации, выбрав нужную.



Скрыть /развернуть фильтры.
Предна значен для отображения /скрытия панели фильтрации. Скрытие /разворачивание панели не влияет на введённые в фильтр данные.



Поиск.
Ключевый элемент



		для панели фильтров, информация по выбранному фильтру не отображается, пока не будет нажат поиск	
	-	Очистка фильтра. Для быстрого сброса всех заданных параметров фильтра	
		Выбрать. Элемент присутствует во вложенных и составных фильтрах. Осуществляет выбор записи из таблицы с отфильтрованной информацией	 
	-	Отмена. Элемент присутствует во вложенных и составных фильтрах. Отменяет выбор	

		записи из таблицы с отфильтрованной информацией	
Светлая тема			Темная тема

Таблица 2 – Панель управления для блоков данных

Активный элемент	Неактивный элемент	Описание элемента	Активный элемент	Неактивный элемент
		Собиратель кнопок (кнопка «еще»). Элемент предназначен для скрытия функциональных кнопок, которые не требуется использовать часто. Для выбора скрытой в собирателе кнопки, требуется нажать на элемент, после чего раскроется список скрытых кнопок, выбрать нужную.		
		Собиратель кнопок для периодов. Элемент предназначен для скрытия обобщения процедур закрытия /открытия расчетных периодов.		
		Добавить. Элемент для добавления данных в выбранном блоке		
	-	Редактировать. Элемент для редактирования данных в выбранном блоке		-
	-	Клонировать. Элемент предназначен для добавления новых данных путём копирования имеющихся данных		-

		Удалить. Элемент для удаления данных в выбранном блоке		
		Информация. Отображает информацию о последнем изменении записи – дату и пользователя системы		
		Обновить. Посыпает запрос на актуализацию записей в базе данных		
НАСТРОЙТЬ	НАСТРОЙТЬ	Вызов мастера заполнения. Присутствует не на всех формах, используется для массовых операций. Функции будут рассмотрены при описании конкретных форм.		
 или Печать в excel		Выгрузка (печать) в Excel. Осуществляет выгрузку представления сформированной в рабочей области таблицы в формате Excel-документа. Для каждой таблицы (блока данных) используется отдельный элемент данного типа.	 или Печать в excel	
Светлая тема				Темная тема

Для удобства представления и работы данные могут быть разбиты по вкладкам, страницам, логическим группам, имеют навигацию по истории изменений, а также возможность сортировок по столбцам.

Таблица 3 – Панели представления информации

Светлая тема	Описание элемента	Темная тема
	Вкладки. Все данные записей сгруппированы по определённым показателям. Для удобства отображения,	

	исчерпыва ющая информац ия отображае тся по группам на вкладках	
	Пагинатор. В случае, если все данные не помещаются на рабочую область, данные разбиваются на страницы, переход по страницам осуществляется путём пролистывания страниц кнопками пагинатора («<>» - переход на предыдущую страницу, «>>» - переход на следующую страницу, «<<>>» - переход к первой странице, «>>>» - переход к последней странице). Также здесь отображается количество страниц и текущее положение.	
	История. Просмотр историчных данных	

		по записи (при наличии)	
ИНН ↑	КПП	Сортировк a. Для выполнения сортировки по необходимому столбцу, необходимо щёлкнуть по заголовку столбца (по возрастанию «а-я» – стрелка вверх, по убыванию «я-а» - стрелка вниз). При отсутствии стрелки, сортировка производится по умолчанию (дате добавления). Столбец, по которому выполнена сортировка, выделяется другим цветом, рядом с наименованием отображается стрелка	ИНН ↑ КПП
		Группиров ки. Используются для удобства отображения информации при наличии	

<p>Реквизиты объекта</p> <p>Количественные характеристики</p> <p>Состояние</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;"> Общий износ здания, % Дата, на которую у... Дата признания дома а... Состояние дома Дата проведения энерг... </div> <p>Конструктивные элементы</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;"> Тип фундамента Форма крыши Материал фундамента Вид несущей части </div>	<p>логических групп информации о полей. По умолчанию все панели развернуты, их можно свернуть нажатием на наименование группы. Разворачивание происходит аналогично</p> <p>Реквизиты объекта</p> <p>Количественные характеристики</p> <p>Состояние</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;"> Общий износ здания, % Дата, на которую у... Дата признания дома а... Состояние дома Дата проведения энерг... </div> <p>Конструктивные элементы</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;"> Тип фундамента Форма крыши Материал фундамента Вид несущей части </div>
--	---

Для удобства работы записи в гридах также имеют элементы управления. Это позволяет осуществлять безошибочное редактирование, просмотр подчинённых записей, быстрый переход к связанным данным на других страницах. Располагаются такие элементы всегда слева в первом столбце, непосредственно напротив каждой записи.

Таблица 4 – Элементы управления для записей

Светлая тема	Описание элемента	Темная тема
	Иерархия. Отображение/скрытие иерархии или дополнительных данных. Если в записи присутствует список дополнительной информации или подчинённых записей, которые можно просмотреть, раскрыть/скрыть список следует нажатием на данный элемент (значок «>» - список скрыт, значок «V» - список раскрыт)	
	Редактирование. Для внесения изменений в выбранную запись	
	Связанные данные. Для быстрого перехода к формам со связанными данными	

Элементы ввода данных

Все элементы ввода данных являются стандартными.

Элементы могут быть обязательными и необязательными для заполнения.

Обязательные для заполнения элементы отмечены красной звёздочкой. Признак обязательности заполнения элемента распространяется на элементы ввода данных любого типа.

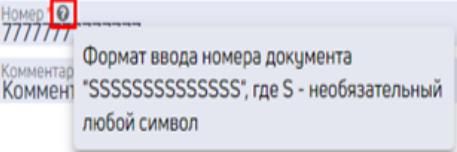
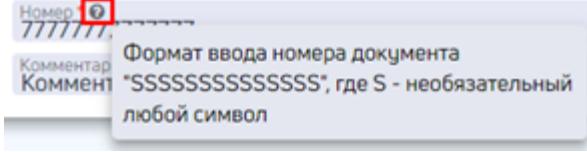
Необязательные поля могут быть заполнены при наличии информации или по мере её поступления.

Элементы также могут быть неактивны, тогда они будут подсвечены другим цветом и заблокированы для заполнения.

Таблица 1 – Элементы ввода данных

Светлая тема	Описание элемента	Темная тема
	Признак обязательности поля	

<p>Организация</p> 	<p>Поле фильтрации. Для внесения данных в такие поля, необходимо нажать на значок лупы и выбрать запись из выпадающего фильтра</p>	<p>Организация</p> 
<p>Наименование</p>	<p>Простое поле. В поле осуществляется ввод текстовой или числовой информации.</p>	<p>Наименование</p>
<p>Период*</p> 	<p>Поле даты. В поле осуществляется ввод даты в заданном формате (например, ДД.ММ.ГГГГ) или из выпадающего календаря при нажатии на значок.</p>	<p>Период*</p> 
<p>Тип</p> 	<p>Выпадающий список. Содержит определённый список данных, из которых нужно выбрать один пункт. В выпадающем списке можно производить поиск по наименованию записи из списка. Поле может быть обязательным</p>	<p>Тип</p> 
<p>УО не указана</p> 	<p>Чекбокс. Атрибут, выполняет логическую функцию «да /нет», в соответствии с подписью</p>	<p>УО не указана</p> 
<p>Адрес</p>	<p>Поле быстрого перехода. Такие поля позволяют осуществить быстрый переход к</p>	<p>Адрес</p>

	другой странице с автоматическим внесением имеющихся в поле данных	
 <p>Номер </p> <p>Формат ввода номера документа "SSSSSSSSSSSS", где S - необязательный любой символ</p> <p>Комментарий</p>	<p>Подсказка. Располагается непосредственно в поле для заполнения. При наведении курсора на значок подсказки появляется информация по правилам заполнения данного поля</p>	 <p>Номер </p> <p>Формат ввода номера документа "SSSSSSSSSSSS", где S - необязательный любой символ</p> <p>Комментарий</p>

Окна для ввода данных

Ввод и редактирование данных для таблиц происходит с помощью всплывающих модальных окон в центре рабочей области, для групп полей – поверх полей заполнения. Принцип заполнения таких окон одинаковый, необходимо внести информацию и нажать кнопку «Сохранить». Расположение кнопки «Сохранить» может быть в правом нижнем или в левом верхнем углу.

Добавление

Организационно-правовая форма *

Полное наименование *

Краткое наименование организации *

Внутреннее наименование *

ИНН *

КПП *

ОГРН

Добавить ещё
СОХРАНИТЬ
ОТМЕНА

Добавление

Организационно-правовая форма *

Полное наименование *

Краткое наименование организации *

Внутреннее наименование *

ИНН *

КПП *

ОГРН

Добавить ещё
СОХРАНИТЬ
ОТМЕНА

СОХРАНИТЬ **ОТМЕНА** Добавление

Реквизиты объекта

Дата начала действия п	Кадастровый номер
Дата окончания действ	Форма собственности
Номер помещения *	Принадлежит общему
Номер подъезда	
Этаж	

Количественные характеристики

Состояние

Состояние помещения	Номер документа пр...
Дата признания помеще	

Реквизиты объекта

Дата начала действия п	Кадастровый номер
Дата окончания действ	Форма собственности
Номер помещения *	Принадлежит общему
Номер подъезда	
Этаж	

Количественные характеристики

Состояние

Состояние помещения	Номер документа
Дата признания помеще	

Мастер ввода данных (песочные часы)

В некоторых случаях, для внесения информации требуется выполнить несколько однотипных действий. Для оптимизации таких процессов используется множественный выбор из мастера ввода данных.

В мастере ввода данных содержатся два окна, слева отображаются доступные для выбора записи, справа – выбранные записи. Для добавления записей, необходимо выбрать запись и перенести в правое окно кнопкой единичного переноса «>». Таким же образом осуществляется снятие выбора с записи, путём нажатия кнопки «<». При необходимости, можно выбрать все доступные записи кнопкой «>>» или снять выбор со всех записей «<<». После выбора необходимых записей, окно следует закрыть при помощи кнопки «Закрыть». Выбранные записи будут добавлены в соответствующий блок.

Настройте Услуги на Объекте обслуживания

Доступные

Дата на...	Дата ок...	Услуга	Услуга по дого...	Поставщик
01.06.2018		Электроснабжение т...	Электроснабжение т...	ИВМ Кратков
01.06.2018		Горячее водоснабже...	Горячее водоснабже...	ЗАО Фрязинская те...

Выбранные

Услуга	Услуга по договору	Поставщик
Холодное водоснабжение	Холодная вода	567-МУП ВОДОКАНАЛ

ЗАКРЫТЬ

Настройте Услуги на Объекте обслуживания

Доступные

Дата ...	Дата ...	Услуга	Услуга п...	Постав...
01.06.2018		Электроснабжени...	Электроснабжени...	IBM Краткоср...
01.06.2018		Горячее водоснаб...	Горячее водоснаб...	ЗАО Фрязинска...

Выбранные

Услуга	Услуга по догов...	Поставщик
Холодное водоснабжение	Холодная вода	567-МУП ВОДОКАНАЛ

» » < «

ЗАКРЫТЬ

Преимущество такого подхода заключается в удобстве представления и добавления информации. Также одна запись не может быть выбрана дважды, что исключает возможность дублирования или пересечения данных.

Компоненты конструктора

В данном каталоге содержатся статьи с описанием всех объектов и основных настроек их атрибутов

- Гриды
 - Простой грид (Grid)
 - Грид типа "Дерево" (Grid Tree)
- Кнопки (Button)
 - Собиратель кнопок (Button Collector)
- Колонки
 - Колонка "Boolean" (Column Boolean)
 - Колонка "Дата" (Column Date)
 - Колонки-кнопки
 - Колонка "Флаг" (Column Checkbox)
 - Колонка-кнопка "Редактирование" (Column Edit)
 - Колонка-кнопка "Удаление" (Column Delete)
 - Колонка-кнопка для отображения значка детализации (Column Detail)
 - Колонка-кнопка с иконкой (Column Action Icon)
 - Текстовые колонки
 - Колонка "Дерево" (Column Tree)
 - Колонка "Строка" (Column String)
 - Числовые колонки
 - Колонка "Целое число" (Column Integer)
 - Колонка "Число" (Column Numeric)
- Окно (Window)
 - Автособираемое окно
 - Кастомное окно
- Панели
 - Динамическая панель (Dynamic Panel)
 - Контейнер типа "Box"
 - Контейнер типа «Панель» (Panel)
 - Панель детализации в гриде (Panel Detail)
 - Панель загрузки и отображения файлов (Panel File)
 - Панель с историчными данными (Panel History)
 - Панель со вкладками (Tab Panel)
 - Панель с фильтрами (Filter Panel)
- Поля
 - Групповые поля
 - Field Group
 - Field Set
 - Поле "Выпадающая таблица" (Field Grid)
 - Поле "Выпадающий список" (Field Combobox)
 - Поле "Изображение" (Field Image)
 - Поле "Песочные часы" (Field Item Selector)
 - Поле "Флаг" (Field Checkbox)
 - Поля с датами

- Поле "Дата" (Field Date)
- Поле "Период" (Field Period Date)
- Пустое пространство (Empty Space)
- Текстовые поля
 - Поле "Многострочный текст" (Field Textarea)
 - Поле "Пароль" (Field Password)
 - Поле "Текст" (Field Text)
 - Расширенный текстовый редактор (Field WYSIWYG)
 - Скрытое поле (Field Hidden)
- Стока в меню "Переход на страницу" (Menu Redirect)
- графики

Гриды

Компоненты для отображения данных в виде таблиц.

- Простой грид (Grid)
- Грид типа "Дерево" (Grid Tree)

Основные настройки для всех типов гридов

Может содержать в себе следующие типы объектов:

- Кнопки, кнопки-колонки, собиратели кнопок;
- Фильтр;
- Колонки (числовые, строковые, с датами и т.д.);
- Поля детализации;
- Переходы на другую страницу;
- Окна.

Кнопки с основными действиями вынесены в атрибуты грида и по умолчанию имеют значение «true»:



1. btndelete – кнопка «Удалить»;
2. btnaudit – кнопка «Информация»;
3. btnexcel – кнопка «Печать в excel»;
4. btnrefresh – кнопка «Обновить».

Сортировка данных в гриде настраивается с помощью атрибутов:

- orderproperty – наименование ключа-значения из сервиса, по которому будет отсортирован грид;
- orderdirection – направление сортировки (ASC/DESC).

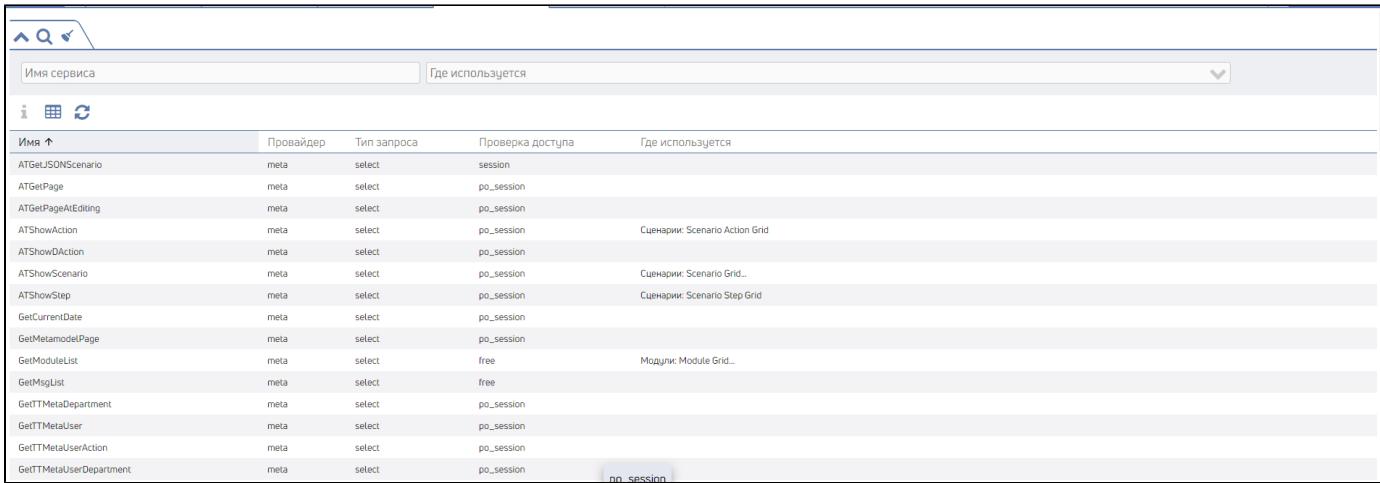
Настройка работы со значениями грида производится с помощью атрибутов:

Наименование атрибута	Описание
collectionvalues	Признак сбора значений: object - виде объекта; array - массив строк.
selmode	Возможность выделения нескольких значений в GRID и TREEGRID SINGLE - только 1 значение; SIMPLE - позволяет легко выбирать значения одно-за-другим. Каждое нажатие добавляет /удаляет значение; MULTI - позволяет комплексно выбирать значения, с учетом ctrl и shift.

С помощью атрибута `splitter` настраивается отображение сплита.

Простой грид (Grid)

Представляет собой таблицу, отображающий данные в строках и столбцах.



The screenshot shows a grid interface with the following columns: Имя (Name), Провайдер (Provider), Тип запроса (Request Type), Проверка доступа (Access Check), and Где используется (Where used). The data rows include ATGetJSONScenario, ATGetPage, ATGetPageAtEditing, ATShowAction, ATShowAction, ATShowScenario, ATShowStep, GetCurrentDate, GetMetamodelPage, GetModuleList, GetMsgList, GetTTMetaDepartment, GetTTMetaUser, GetTTMetaUserAction, and GetTTMetaUserDepartment. Some rows have additional notes like 'Сценарии: Scenario Action Grid...' or 'Модули: Module Grid...'.

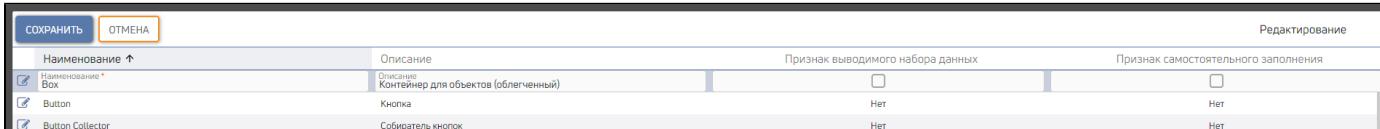
Имя ↑	Провайдер	Тип запроса	Проверка доступа	Где используется
ATGetJSONScenario	meta	select	session	
ATGetPage	meta	select	po_session	
ATGetPageAtEditing	meta	select	po_session	
ATShowAction	meta	select	po_session	Сценарии: Scenario Action Grid...
ATShowAction	meta	select	po_session	
ATShowScenario	meta	select	po_session	Сценарии: Scenario Grid...
ATShowStep	meta	select	po_session	Сценарии: Scenario Step Grid
GetCurrentDate	meta	select	po_session	
GetMetamodelPage	meta	select	po_session	
GetModuleList	meta	select	free	Модули: Module Grid...
GetMsgList	meta	select	free	
GetTTMetaDepartment	meta	select	po_session	
GetTTMetaUser	meta	select	po_session	
GetTTMetaUserAction	meta	select	po_session	
GetTTMetaUserDepartment	meta	select	po_session	no_session

Настройки

Основные настройки для всех типов гридов можно посмотреть [тут](#).

Режим добавления/редактирования данных в гриде определяется атрибутом `edittype` и может принимать значения:

- inline – в строке;



The screenshot shows a dialog for editing grid items. It has tabs for 'СОХРАНИТЬ' (Save) and 'ОТМЕНА' (Cancel). The main table has columns: Наименование (Name), Описание (Description), Признак выводимого набора данных (Flag for displaying the data set), and Признак самостоятельного заполнения (Flag for independent filling). There are three items: Box (Контейнер для объектов (облегченный)), Button (Кнопка), and Button Collector (Собиратель кнопок). The 'Box' item is selected.

Наименование ↑		Описание	Признак выводимого набора данных	Признак самостоятельного заполнения
<input checked="" type="checkbox"/>	Наименование *	Описание Контейнер для объектов (облегченный)	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	Button	Кнопка	Нет	Нет
<input checked="" type="checkbox"/>	Button Collector	Собиратель кнопок	Нет	Нет

- modalwindow – в модальном окне.

Добавление Объекта

Класс *

Сервис

Имя объекта *

Описание объекта *

Порядок сортировки *

Отображаемое имя

Метод модификации

Провайдер данных при модификации

СОХРАНИТЬ
ОТМЕНА

При указании типа редактирования **inline** происходит преобразование колонок грида в поля в соответствии с Таблицей:

Колонка грида	Поле при добавлении/редактировании
Column Boolean	Field Checkbox
Column Date	Field Date
Column Integer	Field Integer
Column Numeric	Field Numeric
Column String	Field Text

В случае, если необходимо, чтобы на основании колонки класса String в автособираемое окно попало не текстовое поле (Field Text), а, например, Поле "Выпадающий список" (Field Combobox), необходимо:

1. выбрать нужную колонку грида;
2. создать дочерний объект класса Combobox;
3. корректно настроить его (указать верные column, displayfield, valuefield, сервис)
4. указать необходимый режим добавления/редактирования в атрибуте editmode.

Возможные значения:

- all - доступно во всех режимах;
- insert - возможен ввод данных только при добавлении;
- update - возможно только редактирование;
- disabled - недоступно для ввода данных.

При указании типа редактирования **modalwindow** и отсутствии объекта класса Окно (Window) в гриде, на основании колонок грида будет сформировано автособираемое окно (подробнее [тут](#)).

Число, указанное в атрибуте pagesize, будет влиять на количество отображаемых строк грида на одной странице.

Также у грида будет отрисован пагинатор, указывающий какая страница из общего количества сейчас отображена.



Грид типа "Дерево" (Grid Tree)

Представляет собой данные, отображаемые в виде иерархической таблицы.

Назначение страницы		Код действия просмотра	Код действия модификации	Порядок ↑
Корневой каталог				1
CORE				1
Конструктор GUI				
Страницы	511	512	1	
Объекты	499	500	2	
Классы	497	498	3	
Атрибуты	491	492	4	
Сервисы	515	516	5	
Модули	515	516	10	
Администрирование				2
Авторизация				3
...				

В гридах типа «Дерево» отсутствуют пагинация и колоночные фильтры.

Для корректного отображения Грид-дерево должен содержать колонку класса – **Колонка "Дерево"** (Column Tree). В данной колонке должны содержаться данные о родительских объектах первого уровня для построения верной иерархии.

Настройки

Основные настройки для всех типов гридов можно посмотреть тут.

Атрибут rootvisible - включает (true) / выключает (false) отображение корневого элемента грида.

Особенности сервиса

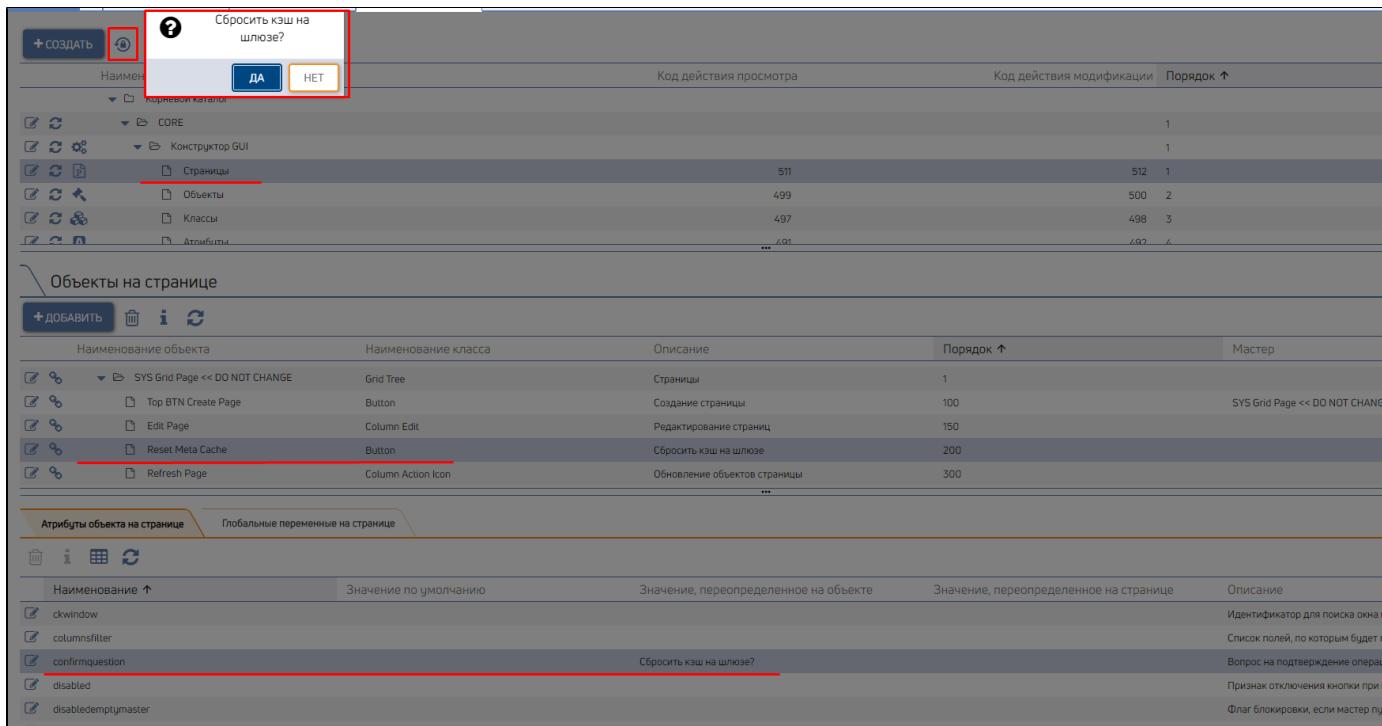
Для корректного отображение иерархической структуры грида типа "Дерево" в результирующем наборе сервиса должны быть следующие ключ-значения:

Наименование ключа	Комментарии
ck_id	Уникальный идентификатор записи
ck_parent	Уникальный идентификатор родительского узла. Пустой для корневого узла.
leaf	Является ли узел листом. true - если узел является листом; false - если узел является корнем.

Кнопки (Button)

Если грид содержит несколько окон, то настроить взаимосвязь конкретной кнопки с конкретным окном позволяет парный атрибут ckwwindow. Для настройки связи необходимо указать одинаковое значение атрибута ckwwindow и у окна, и у кнопки.

Если для выполнения действия после нажатия кнопки необходимо получить подтверждение пользователя, то вопрос-подтверждение указывается в атрибуте confirmquestion.



Для настройки операции, которую будет выполнять кнопка используются атрибуты:

- handler - какой обработчик использовать.

Возможные значения:

Наименование	Описание
onCreateChildWindowMaster	вызывает окно (Window) при добавлении записи в гриде со значением атрибута edittype = modalwindow
onRowCreateChildWindowMaster	вызывает окно (Window) при редактировании записи в гриде со значением атрибута edittype = modalwindow
onSimpleSaveWindow	сохранение данных по кнопке для модального окна
onCloseWindow	закрытие модального окна. После нажатия на кнопку отображается сообщение-подтверждение действия: "Отменить? Да/Нет"
onCloseWindowSilent	закрытие модального окна без сообщения
onPrintHandleOnline	Онлайн печать
onPrintHandleOffline	Отложенная печать

- mode - Тип операции, вызываемой кнопкой

Возможные значения:

Наименование	Описание
1	Добавление
2	Редактирование
3	Удаление
4	Вызов сервиса из атрибута updatequery
5	Вызов меню с информацией
6	Режим клонирования значений
7	Режим скачивания вызов сервиса из атрибута updatequery
8	Режим закачивания файла

- modeaction - Наименование action добавляемое в json вместо стандартных I,U,D
- updatequery - Имя сервиса для кастомных операций
- filetypes - Тип документа доступный для выбора в 8 методе
- maxfile - Максимальный размер файла в байтах

Для настройки иконки кнопки используют атрибуты:

iconfont - Иконка (например, fa-plus)

iconfontname - Наименование класса или наименование колонки при динамике

onlyicon - отображение только иконки

tipmsg - Подсказка у кнопки при наведении

uatype - вид кнопки (1-primary, 2-secondary)

Светлая тема	uatype	Темная тема
	1	
	2	

Отображение кнопок при инициализации настраивается с помощью атрибутов:

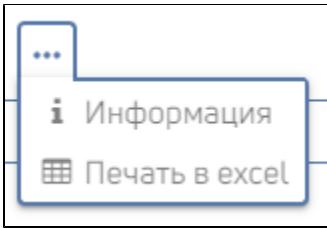
- disabled - признак неактивности кнопки
- disabledemptymaster - признак неактивности кнопки при пустом мастере
- hidden - признак скрытия кнопки

С помощью атрибутов columnsfilter и redirecturl настраивается переход по гиперссылке. Подробнее [тут](#)

Собиратель кнопок (Button Collector)

Позволяет собирать несколько кнопок в один компонент.

При указании в атрибуте btncollectorall значения "true" Собиратель кнопок автоматически включит в себя кнопки Аудит и Печать в Excel.



Колонки

Являются составной частью гридов (простой, грид-дерево, поле "Выпадающая таблица"). Объекты класса Column Action Icon (Колонка-кнопка с иконкой) могут являться самостоятельными и не иметь родителя.

- Колонка "Boolean" (Column Boolean)
- Колонка "Дата" (Column Date)
- Колонки-кнопки
 - Колонка "Флаг" (Column Checkbox)
 - Колонка-кнопка "Редактирование" (Column Edit)
 - Колонка-кнопка "Удаление" (Column Delete)
 - Колонка-кнопка для отображения значка детализации (Column Detail)
 - Колонка-кнопка с иконкой (Column Action Icon)
- Текстовые колонки
 - Колонка "Дерево" (Column Tree)
 - Колонка "Строка" (Column String)
- Числовые колонки
 - Колонка "Целое число" (Column Integer)
 - Колонка "Число" (Column Numeric)

Основные настройки колонок

Для отображения данных в колонках (кроме колонок-кнопок) в атрибуте column следует указать наименование ключа из запроса.

Для настройки местоположения текста в колонке используется атрибут align с доступными значениями:

- left - слева;
- center - по центру;
- right - справа

Значения по умолчанию задаются с помощью атрибутов defaultvalue и defaultvaluequery.

В атрибуте defaultvalue указывается статичное значение. Примеры:

- true/false - для checkbox;
- 1/0 - для boolean;
- sysdate - для указания текущей даты/времени в формате ISO 8601 в Column Date.

В атрибуте defaultvaluequery указывается наименование сервиса, возвращаемое значение которого и будет являться значением по умолчанию для колонки.

Атрибут editmode влияет на доступность ввода данных в колонку в режиме добавления/редактирования и может принимать значения:

- all - доступна во всех режимах;
- insert - возможен ввод данных только при добавлении;
- update - возможно только редактирование;
- disabled - недоступна для ввода данных.

Максимальное количество символов в колонке задается атрибутом maxsize.

В атрибуте regexp указывается регулярное выражение для проверки введенного значения.

Обязательность заполнения колонки регулируется атрибутом required (true/false).

Для сортировки данных в колонке по значению, отличному от атрибута column, следует указать необходимый ключ из сервиса в атрибуте sortcolumn.

Пример использования:

The screenshot shows the 'Grid Object' configuration window. At the top, there's a navigation bar with tabs for 'Классы' (Classes), 'Страницы' (Pages), and 'Объекты' (Objects). Below the navigation bar is a tree view under 'Конструктор GUI' (GUI Designer) with nodes for 'Страницы' (Pages), 'Объекты' (Objects), 'Классы' (Classes), and 'Атрибуты' (Attributes). A table below lists objects with columns for ID, Name, Description, Order, and Master. The 'Объекты на странице' (Objects on page) section contains a table with columns: Наименование объекта (Object name), Наименование класса (Class name), Описание (Description), Порядок ↑ (Order), and Мастер (Master). The 'Атрибуты объекта на странице' (Page object attributes) section shows attributes like 'setglobal', 'sortcolumn', 'stylerules', 'visibleinwindow', and 'visible'. The 'Глобальные переменные на странице' (Page global variables) section is also visible at the bottom.

Видимость колонки в гриде можно настроить, используя атрибут visible (true/false).

Ширина колонки задается в процентах целым числом от 1 до 100 в атрибуте width.

Колонка "Boolean" (Column Boolean)

Колонка для отображения логического типа данных (Да/Нет).

Может принимать значение 1 или 0. Не может быть пустым.

Признак выводимого набора данных
Нет
Нет
Нет
Нет
Да
Нет
Нет

Колонка "Дата" (Column Date)

Используется для отображения дат в формате ISO 8601.

Формат отображения данных регулируется значением атрибута format.

Возможные значения:

- 1 - ГГГГ;
- 2 - МММ ГГГГ;
- 3 - ДД.ММ.ГГГГ;
- 4 - ДД.ММ.ГГГГ ЧЧ:00;
- 5 - ДД.ММ.ГГГГ ЧЧ:МИ;
- 6 - ДД.ММ.ГГГГ ЧЧ:МИ:СС.

Дата начала ↓
01.04.2018
01.01.2018

Колонки-кнопки

- Колонка "Флаг" (Column Checkbox)
- Колонка-кнопка "Редактирование" (Column Edit)
- Колонка-кнопка "Удаление" (Column Delete)
- Колонка-кнопка для отображения значка детализации (Column Detail)
- Колонка-кнопка с иконкой (Column Action Icon)

Колонка "Флаг" (Column Checkbox)

Разновидность колонки-кнопок.

Используется для выбора нескольких записей в гриде.

Услуга

Наименование услуги

Наименование

Автостоянка

Безучетное потребление

Взнос на капитальный ремонт

Видеонаблюдение

Внутризоновая телефонная связь

Вода для домашних животных

Вода для полива

Водоотведение

Колонка-кнопка "Редактирование" (Column Edit)

Разновидность колонки-кнопки.

Отображает кнопку редактирования для каждой записи в гриде.

+ создать	Наименование ↑	Описание
	Chart	Модуль графиков
	Chart Container	Контейнер графиков
	Column Action Icon	Колонка-кнопка с иконкой
	Column Boolean	Колонка "Boolean" (да/нет)
	Column Checkbox	Колонка "Флаг" (для мультивыбора)
	Column Date	Колонка "Дата"
	Column Delete	Колонка-кнопка "Удаление"
	Column Detail	Колонка-кнопка для отображения Key-value
	Column Edit	Колонка-кнопка "Редактирование"
	Column Integer	Колонка "Целое число"

Колонка-кнопка "Удаление" (Column Delete)

Номер ↑
312,00

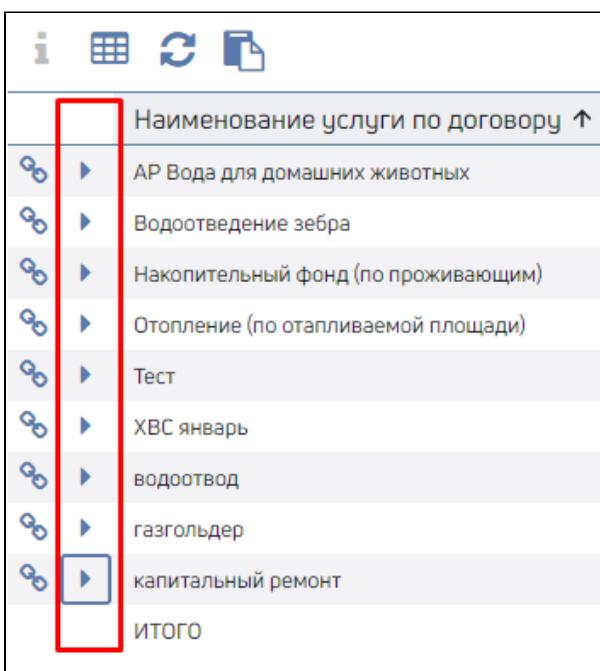
Номер ↑
1.223,00

Разновидность Колонки-кнопки. Используется для удаления выбранной записи грида. Отображается непосредственно в гриде для каждой записи.

Колонка-кнопка для отображения значка детализации (Column Detail)

Разновидность колонки-кнопки.

Используется для настройки видимости значка детализации в гриде. Например, если необходимо скрыть для итоговой строки.



The screenshot shows a grid interface with a toolbar at the top containing icons for information, grid, refresh, and print. The first column contains small blue icons followed by a larger blue icon. The second column contains text labels. A red box highlights the first column, and a blue box highlights the last row, labeled 'ИТОГО'.

Наименование услуги по договору ↑
▶ АР Вода для домашних животных
▶ Водоотведение зебра
▶ Накопительный фонд (по проживающим)
▶ Отопление (по отапливаемой площади)
▶ Тест
▶ ХВС январь
▶ водоотвод
▶ газгольдер
▶ капитальный ремонт
ИТОГО

Колонка-кнопка с иконкой (Column Action Icon)

Если необходимо поместить кнопку для каждой записи в гриде, используется объект класса Колонка-кнопка с иконкой (Column Action Icon).

Данный объект включает в себя все настройки класса Кнопка (ссылка), но является колонкой в гриде.

Примеры использования:

Колонка-кнопка "Обновить привязку" и Кнопка-колонка "Иконка" на странице Страницы:

Наименование страницы

Код действия

- +Создать
- Сохранить
- Удалить
- Информация
- Обновить

Наименование страницы

Код действия

Корневой каталог

CORE

Конструктор GUI

- Страницы
- Объекты
- Классы
- Атрибуты
- Сервисы
- Модули
- Администрирование
- Авторизация

Для отображения разных иконок в зависимости от данных необходимо указать значение "true" для атрибута dynamicicon.

Текстовые колонки

- Колонка "Дерево" (Column Tree)
- Колонка "Строка" (Column String)

Колонка "Дерево" (Column Tree)

Колонка для отображения данных 1 уровня в гриде типа "Дерево".

Наименование страницы	Код действия просмотра	Код действия модификации	Порядок ↑
Корневой каталог			1
CORE			1
Конструктор GUI			
Страницы	511	512	1
Объекты	499	500	2
Классы	497	498	3
Атрибуты	491	492	4
Сервисы	515	516	5
Модули	515	516	10
...			

Объекты на странице

+ Добавить

Наименование объекта

Наименование класса

Описание

Порядок ↑

Мастер

SYS Grid Page << DO NOT CHANGE	Grid Tree	Страницы	1	
Top BTN Create Page	Button	Создание страницы	100	SYS Grid Page <<
Edit Pag	Column Edit	Редактирование страниц	150	
Reset Meta Cache	Button	Сбросить кэш на шлюзе	200	
Refresh Page	Column Action Icon	Обновление объектов страницы	300	
Column Icon Page	Column Action Icon	Колонка выводящая иконку	400	
Page Name	Column Tree	Наименование страницы	500	
cr_type_invisible	Column Integer	cr_type_invisible	550	

Колонка "Строка" (Column String)

Колонка для отображения текстовых данных.

Описание

Колонка "Флаг" (для мультивыбора)

Колонка "Дата"

Колонка-кнопка "Удаление"

Колонка-кнопка для отображения Key-value

Колонка-кнопка "Редактирование"

Колонка "Целое число"

Колонка "Число"

Колонка "Строка"

Колонка "Дерево"

В атрибуте imask указывается маска ввода данных.

Например, (999) 999-99-99

С помощью атрибутов columnsfilter и redirecturl настраивается переход по гиперссылке. Подробнее [тут](#)

Числовые колонки

- Колонка "Целое число" (Column Integer)
- Колонка "Число" (Column Numeric)

Колонка "Целое число" (Column Integer)

Колонка для отображения целочисленных значений.

Максимальное количество символов указывается в атрибуте maxsize.

Максимальное и минимальное значения колонки указываются в атрибутах maxvalue и minvalue соответственно.

Колонка "Число" (Column Numeric)

Колонка для отображения числовых данных.

Номер ↑ ↓
312,00
1.223,00

В атрибуте currenciesign указывается постфикс, который будет добавлен к данным. Например. "руб." для отображения суммы.

В атрибуте decimalprecision указывается целое число, определяющее точность после запятой.

В атрибуте decimalseparator - разделитель дробной части.

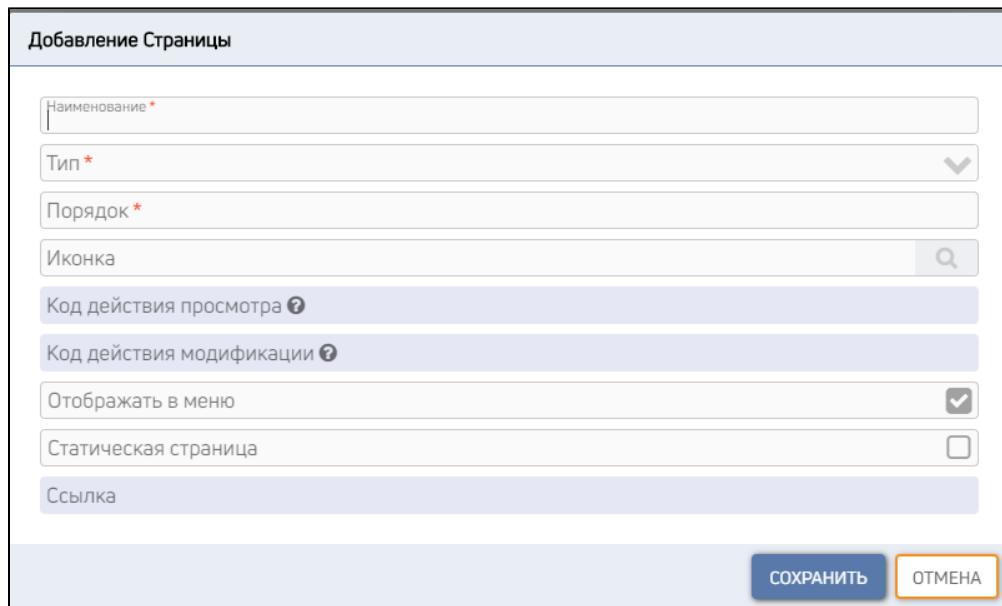
В атрибуте thousandseparator - разделитель тысяч.

Важно!

Значения атрибутов thousandseparator и decimalseparator не могут совпадать.

При попытке переопределения значений указанных атрибутов на объекте и/или странице, которое приведет к совпадению значений, система вернет ошибку: "Значения атрибутов decimalseparator и thousandseparator не могут совпадать"

Окно (Window)



Добавление Страницы

Наименование *

Тип *

Порядок *

Иконка

Код действия просмотра ⓘ

Код действия модификации ⓘ

Отображать в меню

Статическая страница

Ссылка

СОХРАНИТЬ ОТМЕНА

Объект класса Okno может включать в себя объекты классов:

- кнопки и собиратели кнопок;
- панели (Контейнеры типа "Панель" и "Box", Динамическая панель);
- поля;
- гриды (простой и грид-дерево).

Окна используются для добавления/редактирования записей грида.

Различают 2 типа окон:

1. Кастомное (созданный объект класса Window, включающий в себя поля и кнопки);
2. Автособираемое (созданное автоматически на основании колонок грида).

Автособираемое окно

Формируется автоматически на основе колонок грида в случае, если в гриде в edittype указано значение "modalwindow" и выполняется одно из следующих условий:

- в гриде отсутствует объект класса Окно (Window);
- в гриде присутствует объект класса Окно (Window), но значение парного атрибута ckwindow у окна и кнопки (Button) в гриде не совпадает.

- Преобразование колонок грида в поля в автособираемом окне
- Настройка автособираемого окна
- Наименование автособираемого окна

Пример автособираемого окна можно посмотреть на странице [указать страницу!!!!](#)

Преобразование колонок грида в поля в автособираемом окне

При формировании автособираемого окна на основании колонок грида будут сформированы поля в соответствии с таблицей:

Колонка грида	Поле в автособираемом окне
Column Boolean	Field Checkbox
Column Date	Field Date
Column Integer	Field Integer
Column Numeric	Field Numeric
Column String	Field Text

Настройка автособираемого окна

В случае, если необходимо, чтобы на основании колонки класса String в автособираемое окно попало не текстовое поле (Field Text), а, например, Поле "Выпадающий список" (Field Combobox), необходимо:

1. выбрать нужную колонку грида;
2. создать дочерний объект класса Combobox;
3. корректно настроить его (указать верные column, displayfield, valuefield, сервис)
4. указать необходимый режим добавления/редактирования в атрибуте editmode.

Возможные значения:

- all - доступно во всех режимах;
- insert - возможен ввод данных только при добавлении;
- update - возможно только редактирование;
- disabled - недоступно для ввода данных.

При указании значения disabled поле не будет отображаться в модальном окне.

Если необходимо, чтобы поле все-таки отображалось в модальном окне, но было недоступно для ввода /редактирования, необходимо указать значение "true" в атрибуте visibleinwindow.

5.

Наименование автособираемого окна

Наименование модального окна формируется в соответствии с типом модификации кнопки (анализируется значение атрибута mode: 1 - добавление/ 2 - редактирование/ 6 - клонирование), его вызывающей, и значения поля "Описание объекта" соответствующего родительского объекта (грид, грид-дерево, ПИД).

Пример:

1. Описание грида - "Договора"

Редактирование Объекта

Класс*
Grid (Грид для отображения данных)

Сервис
ConShowContracts

Имя объекта*
Contracts Grid

Описание объекта*
Договора

Порядок сортировки*
1500

Отображаемое имя

Метод модификации
pkg_json.con._modify_con

Провайдер данных при модификации*
БФЛ ЖКХ

СОХРАНИТЬ **ОТМЕНА**

2. значение атрибута mode = 1 для кнопки, вызывающей автособираемое окно

Наименование	Значение по умолчанию	Значение (переопределено)
handler		
hidden		
iconfont		fa-plus
iconfontname	fa	
maxfile	5242880	
mode	1	
modeaction		

3. Наименование автособираемого окна: "Добавление Договора"

Добавление Договора

Дата начала*
.

Дата окончания

Номер *

Дата заключения*

Наименование

Подразделение*

Предмет

Тип*

Кастомное окно

Если необходимо завести сложную логику при добавлении/редактировании данных в гриде (например, скрывать поля в определенных случаях), необходимо сформировать кастомное окно.

В объекте класса Окно (Window) создаются необходимые поля и кнопки.

- Настройки
- Наименование кастомного окна
 - Способ 1
 - Способ 2

Настройки

Для настройки связи окна и кнопки его вызывающей используется атрибут ckwindow. Значение данного атрибута для кнопки и окна должно совпадать.

Для отображения в окне чек-бокса "Добавить еще" необходимо указать значение атрибута checkaddmore = true.

Для настройки размеров окна используются следующие атрибуты:

Наименование атрибута	Описание
height	Статическая высота в пикселях (px)
maxheight	Максимальная высота
minheight	Минимальная высота
wintype	Тип окна (влияет на ширину окна): narrow: 500px, default: 800px, wide: 1000px, xwide: 1200px, xlwide: 1600px

Наименование кастомного окна

Способ 1

Если у кастомного окна (Window) заполнено поле "Отображаемое имя", то наименование модального окна соответствует значению данного поля.

Данный способ следует использовать в случае, когда окно используется только для одного типа операции (например, только добавление данных).

Пример:

Окно "Clone Object Window" в гриде "SYS Grid Object << DO NOT CHANGE":

Редактирование Объекта

Класс Window (Окно)
Сервис
Имя объекта* Clone Object Window
Описание объекта* Окно клонирования объекта
Порядок сортировки* 250
Отображаемое имя Клонирование объекта
Метод модификации
Провайдер данных при модификации
СОХРАНИТЬ ОТМЕНА

Клонирование объекта

Родительский объект SYS Grid Object << DO NOT CHANGE	🔍
КЛОНРОВАТЬ ОТМЕНА	

Способ 2

Если поле "Отображаемое имя" не заполнено, то наименование формируется следующим образом: тип модификации кнопки + значение поля "Описание объекта" окна (Window).

Данный способ наименования используется в случае, когда одно модальное окно вызывается для разных модификаций (например, Добавления и редактирования).

Пример:

Окно "Add/Edit Object Window" в гриде "SYS Grid Object << DO NOT CHANGE":

Редактирование Объекта

Класс Window (Окно)
Сервис
Имя объекта* Add/Edit Object Window
Описание объекта* Объекта
Порядок сортировки* 180
Отображаемое имя
Метод модификации
Провайдер данных при модификации
СОХРАНИТЬ ОТМЕНА

Добавление Объекта

Класс *	Сервис
Имя объекта *	Описание объекта *
Порядок сортировки *	Отображаемое имя
Метод модификации	Провайдер данных при модификации

СОХРАНИТЬ **ОТМЕНА**

Редактирование Объекта

Класс Window (Окно)	Сервис
Имя объекта * Add/Edit Object Window	Описание объекта * Объекта
Порядок сортировки * 180	Отображаемое имя
Метод модификации	Провайдер данных при модификации

СОХРАНИТЬ **ОТМЕНА**

Панели

Компоненты типа "Контейнер" для включения в себя других элементов.

- Динамическая панель (Dynamic Panel)
- Контейнер типа "Box"
- Контейнер типа «Панель» (Panel)
- Панель детализации в гриде (Panel Detail)
- Панель загрузки и отображения файлов (Panel File)
- Панель с историческими данными (Panel History)
- Панель со вкладками (Tab Panel)
- Панель с фильтрами (Filter Panel)

Атрибут `contentview` указывает вид наполнения Панели.

Возможные значения:

- `hbox` - горизонтальное (Объекты будут расположены горизонтально друг за другом);
- `hbox-wrap` - горизонтальное с переносом на следующую строку (Если суммарная ширина объектов будет превышать 100% (складываются переопределенные значения атрибутов `width`), то объекты будут перенесены на следующую строку);
- `vbox` - вертикальное (объекты будут расположены вертикально друг под другом).

Динамическая панель (Dynamic Panel)

Представляет собой контейнер, в котором отображаются элементы, возвращаемые с бэка.

Данный компонент реализован для отрисовки:

- на одной странице грида с разным набором колонок (количество/наименование и т.д.).
- в одном модальном окне разного набора полей.

Пример: Имеем многотарифный прибор учета электроэнергии. Кол-во тарифов получаем как атрибут ПУ из БД. Задача создать форму ввода показаний для таких приборов учета.

Создаем процедуру, результатом работы которой будет JSON, описывающий нашу форму ввода.

```
--  
vc_json := '{  
    "cl_dataset": "1",  
    "cv_name": "Dynamic Panel",  
    "ck_page": "' || to_char(pk_page) || '", --  
    "ck_modify": "modify",  
    "cv_helper_color": "red",  
    "contentview": "vbox",  
    "reqsel": "true",  
    "type": "FIELDSET",  
    "datatype": "array",  
    "column": "ca_value",  
    "child": [  
  
        -- child ( )  
        --  
        for vcur_child in (select rownum as cn_cnt,  
                            i.ck_id,  
                            s.cv_short,  
                            s.cr_indication,  
                            case i.cn_value  
                                when null then  
                                    ''  
                                else  
                                    trim(to_char(i.cn_value, rpad(lpad('D',  
vn_capacity_int +1,'9')),  
vn_capacity_int + vn_capacity_dec + 1,  
                                         '9')),  
                                         'NLS_NUMERIC_CHARACTERS =  
                                         '.',','))  
            end as cv_value,  
            i.cl_overstep,  
            decode(nvl(i.cl_overstep, 0), 0, '"checked": "  
0", ', '"defaultvalue": "1", ') cl_overstep_bool  
            from t_indication i  
            left join t_d_service s  
                on s.ck_id = i.ck_d_service  
                where i.cn_key_group = pn_key_group) loop  
        --      EDIT CHECKBOX 0 .  
        if vcur_child.cn_cnt > 1 then  
            vc_json := vc_json || ',';  
        end if;
```

```

vc_json := vc_json || '{
    "cl_dataset": "1",
    "cv_name": "Dynamic Panel",
    "ck_page": "' || to_char(pk_page) || '',
    "ck_modify": "modify",
    "cv_helper_color": "red",
    "contentview": "column",
    "reqsel": "true",
    "type": "FIELDSET",
    "datatype": "array",
    "column": "' || to_char(vcur_child.ck_id) || '',
    "childs": [
        {
            "cl_dataset": "0",
            "cv_name": "cn_value_'" || to_char(vcur_child.cn_cnt) || '',
            "column": "cn_value",
            "cv_displayed": "/ ' || vcur_child.cv_short || '',
            "ck_page": "' || pk_page || '',
            "width": "60%", -- %
            "datatype": "numeric", --
            "decimalprecision": "' || to_char(vn_capacity_dec) || '',
            "thousandseparator": " ", --
            "maxvalue": "' || vv_maxvalue || '',
            "defaultvalue": "' || to_char(vcur_child.cv_value) || '',
            --
            "info": null,
            "required": "true",
            "type": "IFIELD",
            "visibleinwindow": "false"
        },
        {
            "cl_dataset": "0",
            "cv_displayed": " \"\",
            "cv_description": " \"\",
            "ck_page": "' || to_char(pk_page) || '',
            "column": "cl_overstep",
            "fieldlabel": "&nbsp;",
            "width": "40%",
            "datatype": "CHECKBOX", ' || to_char(vcur_child.cl_overstep_bool) ||
            ' "type": "IFIELD",
            "visibleinwindow": "false"}]}' ;
end loop;
vc_json := vc_json || ']}' ;

```

В итоге с бэка должен приходить JSON вида:

```
{
    "success": true,
    "data": [
        {

```

```
        "cl_dataset": "1",
        "cv_name": "Dynamic Panel",
        "ck_page": "301",
        "ck_modify": "modify",
        "cv_helper_color": "red",
        "contentview": "vbox",
        "reqsel": "true",
        "type": "FIELDSET",
        "datatype": "array",
        "column": "ca_value",
        "child": [
            {
                "cl_dataset": "1",
                "cv_name": "Dynamic Panel",
                "ck_page": "301",
                "ck_modify": "modify",
                "cv_helper_color": "red",
                "contentview": "column",
                "reqsel": "true",
                "type": "FIELDSET",
                "datatype": "array",
                "column": "13",
                "child": [
                    {
                        "cl_dataset": "0",
                        "cv_name": "cn_value_1",
                        "column": "cn_value",
                        "cv_displayed": "/",
                        "ck_page": "301",
                        "width": "60%",
                        "datatype": "numeric",
                        "decimalprecision": "9",
                        "thousandseparator": " ",
                        "maxvalue": "99999.99999999",
                        "defaultvalue": "",
                        "info": null,
                        "required": "true",
                        "type": "IFIELD",
                        "visibleinwindow": "false"
                    },
                    {
                        "cl_dataset": "0",
                        "cv_displayed": "\\\\"\\",
                        "cv_description": ""
                    }
                ]
            }
        ]
    }
}
```

```

        " " \ " \" " ,
        "ck_page": "301",
        "column": "fieldlabel": "width": "40%", "datatype": "checked": "0", "type": "FIELD",
        "checkbox": "visibleinwindow": "false"
    }
]
}
]
],
"metaData": {
    "responseTime": 0.029,
    "total": 1
}
}
}

```

Такой JSON после обработки на среднем слое превратится вот в такую форму ввода

Показание/Расход Э/Э 3-тар п/пик*	<input type="text" value="2"/>	Переход через "ноль" <input type="checkbox"/>
Показание/Расход Э/Э 3-тарифн ночь*	<input type="text" value="3"/>	Переход через "ноль" <input type="checkbox"/>
Показание/Расход Э/Э 3-тар пик*	<input type="text" value="78"/> <input type="button" value="X"/>	Переход через "ноль" <input checked="" type="checkbox"/>

Контейнер типа "Box"

Представляет собой облегченную версию контейнера типа "Панель".

Используется для группировки полей в блоки данных.

Реквизиты организации

Банковские реквизиты

Адреса

Роли

Дата начала действия реквизитов *
01.12.2019

ФИО руководителя *
Иванов Иван Иванович

Фамилия руководителя *
Иванов

Имя руководителя *
Иван

Отчество руководителя
Иванович

Документ, удостоверяющий личность
Свидетельство о предоставлении временного убежища на территории Российской Федерации

Серия *
КК

Номер *
1212121

Комментарий

ФИО контактного лица

Телефон организации
(111) 111-11-11

Тип телефона
Городской

Электронная почта организации

Факс организации
22222222

Сайт организации

Телефон дистанционной службы
(333) 333-33-33

Режим работы

Контейнер типа «Панель» (Panel)

Компонент представляет собой контейнер, в который можно поместить практически любые объекты:

- Кнопки и Собиратели кнопок;
 - Фильтры;
 - Гриды (простые и гриды-дерево);
 - Поля (текстовые, числовые и т.д.);
 - Панели (Панели со вкладками, ПИД, контейнеры типа «Панель» и т.д.).

Данный компонент используется для агрегации информации из нескольких типов объектов Конструктора.

Атрибут collapsed определяет, свёрнута ли панель при инициализации страницы (true - свёрнута).

Атрибут `collapsible` указывает, возможно ли свёртывание панели (true/false).

Пример:

Реквизиты объекта
Количественные характеристики
Состояние
Общий износ здания, %
Дата признания дома аварийным
Дата проведения энергетического обследования
Дата, на которую установлен износ здания
Состояние дома
Конструктивные элементы

Панель "Состояние" - развернута, остальные - свёрнуты.

Заблокировать возможность редактирования дочерних объектов панели можно с помощью атрибута `editmodepanel` (Работает в случае, когда панель содержит поля/панели с полями и кнопки (кнопки, содержащиеся в панели будут скрыты). Не влияет на поведение гридов в панелях).

Панель детализации в гриде (Panel Detail)

Позволяет выводить поля детализации к записи в гриде.

Наименование услуги по договору ↑	Признак ОДН	Сальдо на начало месяца	Начислено	Ручная корректировка	Оплачено
AP Вода для домашних животных					
Поставщик ООО "АРИАЛ"	Договор 1-1		Подразделение КО "Соколиная гора"		
▶ Водоотведение зебра		333.239,40	1.066,74		
▶ Накопительный фонд (по проживающим)		38,55	0,91		
▶ Отопление (по отапливаемой площади)		640,03	16,52		

Атрибут contentview указывает вид наполнения Панели.

Возможные значения:

- **hbox** - горизонтальное (Объекты будут расположены горизонтально друг за другом);

- hbox-wrap - горизонтальное с переносом на следующую строку (Если суммарная ширина объектов будет превышать 100% (складываются переопределенные значения атрибутов width), то объекты будут перенесены на следующую строку);
- vbox - вертикальное (объекты будут расположены вертикально друг под другом).

Панель загрузки и отображения файлов (Panel File)

Компонент для загрузки и отображения загруженных файлов.

Для отображения наименования файлов необходимо, чтобы сервис возвращал ключ-значение cv_file_name.

Лейбл собирается из ключ-значений: cv_dd_name + cv_number + cd_date.

Также Panel File имеет:

"Override Add Button"
 "Override Cancel Button"
 "Override Delete Button"
 "Override Download Button"
 "Override Save Button"

Панель с историчными данными (Panel History)

Компонент реализован для отображения исторических данных (атрибуты, реквизиты и т.д.) Представляет собой панель с определенным набором полей и управляющим элементом.

The screenshot shows a user interface for managing historical data. At the top, there are tabs: 'Паспорт дома', 'Способ формирования ФКР', 'Договоры по объекту обслуживания', and 'Документы'. Below the tabs is a toolbar with icons for add (+), edit (pencil), clone (copy), delete (trash), refresh, back, forward, and information (i). The main area is divided into sections by orange horizontal bars. The first section, 'Реквизиты объекта', contains fields: 'Дата начала действия паспорта' (10.05.2018), 'Класс энергетической эффективности'; 'Год ввода в эксплуатацию', 'Код ОКТМО'; 'Год постройки', 'Номер дома в ГИС ЖКХ'; 'Год реконструкции', 'Серия дома'; 'Кадастровый номер', 'Код УСЭН'; 'Имеются приспособления для нужд МГН', 'Часовой пояс' (checkbox checked); 'Является объектом культурного наследия', 'Является домом блокированной застройки' (checkbox unchecked). The second section, 'Количественные характеристики', contains fields: 'Общий износ здания, %', 'Дата, на которую установлен износ здания'; 'Дата признания дома аварийным', 'Состояние дома'; 'Дата проведения энергетического обследования'. The third section, 'Конструктивные элементы', is partially visible at the bottom.

Управляющим элементом содержит 8 кнопок:



1. Добавить;
2. Редактировать;
3. Клонировать;
4. Удалить;
5. Предыдущая запись;
6. Следующая запись;
7. Информация.

По умолчанию отображаются данные на максимальную дату. При наличии записей на предыдущие даты переключение между их отображением осуществляется с помощью кнопок навигации (Предыдущая запись, Следующая запись).

При создании новой записи, отличающейся от предыдущей лишь несколькими полями, следует использовать кнопку «Клонировать». В этом случае все поля ПИД заполняются значениями из предыдущей записи кроме даты самой записи (например, даты начала действия паспорта объекта обслуживания). Пользователь указывает новую дату записи, изменяет значения нужных полей и сохраняет новую запись. Данная функция позволяет избежать ручного набора повторяющихся значений.

Общие правила ведения исторических данных

Существует 2 вида ведения истории изменения атрибутов:

1. Без даты окончания (Дата окончания действия атрибутов не ведется пользователем, в БД Дата окончания действия предусмотрена):
 - при **добавлении** новой записи необходимо проверить, что предыдущая запись закрыта и мы вставляем новую запись;
 - при **изменении** данных мы должны проверить, что новая дата вставляемой записи не пересекается с предыдущим(-ими) интервалом(-ами);
 - у предыдущей записи (и при **добавлении** новой записи, и при **редактировании**) необходимо выставить Дату окончания = Дата начала вставляемой записи -1 секунда;
 - при **удалении** последней записи (на max(Дата начала)) у предыдущей записи проставлять Дата окончания = null.
2. С датой окончания (Дата окончания действия атрибутов ведется пользователем):
 - при добавлении новой записи необходимо проверить, что у предыдущей записи заполнена Дата окончания действия и она меньше Даты начала действия вставляемой записи;
 - при редактировании/изменении необходимо проверить, что редактируем последнюю (с max(Дата начала действия)) запись и Дата окончания действия предыдущей записи меньше Даты начала обновляемой; к Дате окончания предыдущей записи прибавляем период 23 часа 59 минут 59 секунд.

Проверки со стороны БД

Без даты окончания

```
/*
for pcur_max in (select a.cd_st cd_st_max,
a.ck_id ck_id_max,
lead(cd_st) over(order by cd_st desc) cd_st_prev,
lead(ck_id) over(order by cd_st desc) ck_id_prev
from t_org_attribute a
where a.ck_org = pot_org_attribute.ck_org
order by cd_st desc fetch next 1 rows only) loop
vd_st_max := cd_st_max;
vk_id_max := ck_id_max;
vd_st_prev := cd_st_prev;
vk_id_prev := ck_id_prev;
--
if vd_st_max >= pot_org_attribute.cd_st and pv_action = pkg.I then
pkg.p_set_error(1131); /* */
end if;
if pot_org_attribute.ck_id != vk_id_max and pv_action = pkg.U then
pkg.p_set_error(1132); /* */
end if;
if vd_st_prev >= pot_org_attribute.cd_st and pot_org_attribute.ck_id =
vk_id_max and pv_action = pkg.U then
pkg.p_set_error(1133); /* */
end if;
end loop;
```

```

if pv_action = pkg.I and vv_error is null then
/* " " */
pot_org_attribute.ck_id := seq_org_attribute.nextval;
insert into t_org_attribute values pot_org_attribute;
--
update t_org_attribute a set cd_en=pot_org_attribute.cd_st-interval '1'
second where ck_id = vk_id_max;
elsif pv_action = pkg.U and vv_error is null then
/* " " */
update t_org_attribute
set row = pot_org_attribute
where ck_id = pot_org_attribute.ck_id;
--
update t_org_attribute a set cd_en=pot_org_attribute.cd_st-interval '1'
second where ck_id = vk_id_prev;
if sql%notfound then
pkg.p_set_error(504);
end if;
end if;

```

С датой окончания

```

/* ,      */
for pcur_max in (select a.cd_en cd_en_max,
a.ck_id ck_id_max,
lead(cd_en) over(order by cd_st desc) cd_en_prev
from t_org_attribute a
where a.ck_org = pot_org_attribute.ck_org
order by cd_st desc fetch next 1 rows only) loop
vd_en_max:= cd_en_max;
vk_id_max:= ck_id_max;
vd_en_prev:= cd_en_prev;
--
if vd_en_max is null and pv_action = pkg.I then
pkg.p_set_error(1134); /*      */
end if;
if vd_en_max >= pot_org_attribute.cd_st and pv_action = pkg.I then
pkg.p_set_error(1135); /* ,      */
end if;
if pot_org_attribute.ck_id != vk_id_max and pv_action = pkg.U then
pkg.p_set_error(1132); /*      */
end if;
if vd_en_prev >= pot_org_attribute.cd_st and pot_org_attribute.ck_id =
vk_id_max and pv_action = pkg.U then
pkg.p_set_error(1136); /* ,      */
end if;
end loop;
if pot_org_attribute.cd_en is not null then

```

```
pot_org_attribute.cd_en:=trunc(pot_org_attribute.cd_en,'DD')+to_dsinterval
('0 23:59:59');
end if;
```

Панель со вкладками (Tab Panel)

Компонент Панель со вкладками реализован для отображения на одной странице нескольких вкладок с данными.



Панель со вкладками может включать в себя следующие типы объектов:

- Гриды (Простой грид, Грид-дерево);
- Контейнер типа «Панель»;
- Панель с историческими данными (ПИД);
- Панель со вкладками.

При добавлении объекта в Tab Panel он будет отображаться в новой вкладке с именем, соответствующий значению поля «Отображаемое имя» добавляемого объекта.

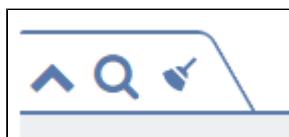
Если необходимо разместить несколько объектов на одной вкладке, то в Панели со вкладками создается Контейнер вида "Панель" (Panel), в котором уже создаются нужные объекты.

Панель с фильтрами (Filter Panel)

Представляет панель с полями, по которым будет происходить фильтрация.



При значении атрибута dynamicfilter = true включается "динамический режим" панели фильтров: при вводе данных в любое поле фильтра происходит автоматическая перезагрузка грида.



Кроме того, убирается панель управления фильтра:



По умолчанию данные, по которым осуществлялся последний поиск, сохраняются в кэше. Т.е. при повторном открытии закрытой страницы с фильтром, в поля фильтрации будут проставлены ранее введенные значения.

Отключить сохранение фильтров можно с помощью атрибута filtervaluesave = false.

Поля

Дочерние элементы для контейнеров, окон и некоторых типов колонок.

- Групповые поля
 - Field Group
 - Field Set

- Поле "Выпадающая таблица" (Field Grid)
 - Поле "Выпадающий список" (Field Combobox)
 - Поле "Изображение" (Field Image)
 - Поле "Песочные часы" (Field Item Selector)
 - Поле "Флаг" (Field Checkbox)
 - Поля с датами
 - Поле "Дата" (Field Date)
 - Поле "Период" (Field Period Date)
 - Пустое пространство (Empty Space)
 - Текстовые поля
 - Поле "Многострочный текст" (Field Textarea)
 - Поле "Пароль" (Field Password)
 - Поле "Текст" (Field Text)
 - Расширенный текстовый редактор (Field WYSIWYG)
 - Скрытое поле (Field Hidden)
-

Основные настройки полей

Для отображения данных в полях (кроме групповых полей: Field Set и Field Group) в атрибуте column следует указать наименование ключа из запроса. Для Поля "Период" (Field Period Date) используются атрибуты columnstart и columnend для задания начала и конца периода.

Значения по умолчанию задаются с помощью атрибутов defaultvalue и defaultvaluequery.

В атрибуте defaultvalue указывается статичное значение. Примеры:

- true/false - для checkbox;
- 1/0 - для boolean;
- sysdate - для указания текущей даты/времени в формате ISO 8601 в Column Date;
- first - для выбора первой записи (Field Grid, Field Combobox);
- alwaysfirst - для выбора первой записи всегда (Field Grid, Field Combobox).

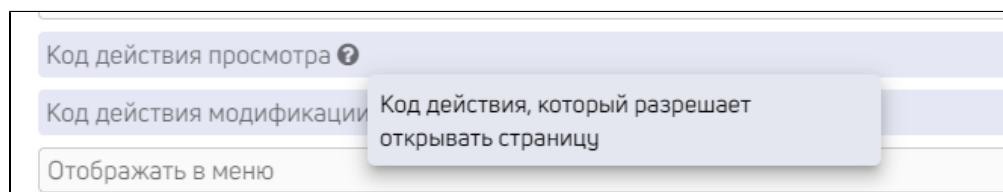
В атрибуте defaultvaluequery указывается наименование сервиса, возвращаемое значение которого и будет являться значением по умолчанию для поля.

Атрибут editmode влияет на доступность ввода данных в поле в режиме добавления/редактирования и может принимать значения:

- all - доступна во всех режимах;
 - insert - возможен ввод данных только при добавлении;
 - update - возможно только редактирование;
 - disabled - недоступна для ввода данных.
-

Для добавления дополнительной информации по полю используется атрибут info. При наличии значения данного атрибута у поля будет отображаться значок "?".

При наведении курсора мыши на указанный значок, будет отображаться значение атрибута info.



Максимальное количество символов в поле задается атрибутом maxsize.

В атрибуте regexp указывается регулярное выражение для проверки введенного значения.

Обязательность заполнения поля регулируется атрибутом required (true/false).

Групповые поля

Компоненты для визуального (Field Group) или логического (Field Set) объединения полей в одну группу.

Field Group

Field Set

Поле "Выпадающая таблица" (Field Grid)

Позволяет в поле выбирать одно значение из грида.

Основные настройки

Может содержать в себе:

- колонки;
- панель детализации;
- фильтр.

Сортировка данных в гриде настраивается с помощью атрибутов:

- orderproperty – наименование ключа-значения из сервиса, по которому будет отсортирован грид;
- orderdirection – направление сортировки (ASC/DESC).

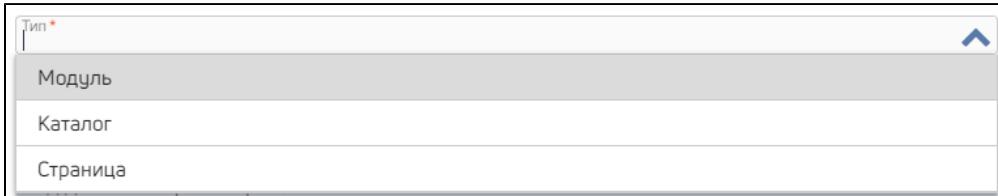
В атрибуте column указывается ключ-значение, в которое будет передаваться ck_id выбранной в гриде записи.

Настройка размера выпадающего грида настраивается с помощью атрибутов:

Наименование атрибута	Описание
pickerheight	Высота выпадающего меню для выбора. По умолчанию 390px
pickerwidth	Ширина выпадающего меню для выбора

Поле "Выпадающий список" (Field Combobox)

Данный компонент позволяет пользователю выбрать одно значение из списка предложенных или при необходимости ввести новое значение.



Для корректной работы поля необходимо указать сервис, возвращающий данные в combobox и верно настроить атрибуты:

Наименование атрибута	Описание	Комментарии

column	Наименование колонки в запросе, из которой берутся данные	ключ-значение, которое будет передано в сервис
displayfield	Имя параметра, который будет отображен	ключ-значение, которое будет отображено в интерфейсе
valuefield	Поле содержит информацию о файле	ключ-значение из сервиса combobox, из которого берется значение

С помощью атрибута allownew включается/выключается (true/false) возможность введения нового значения через combobox.

Настройка фильтрации значений в combobox настраивается с помощью атрибутов:

Наименование атрибута	Описание	Комментарии
minchars	Количество введенных символов для получения подсказок	сколько символов необходимо ввести, чтобы отработала фильтрация значений combobox.
querydelay	Время паузы перед вызовом сервиса с указанием фильтра	
querymode	Работа с combobox: remote или local	remote - сервис каждый раз запрашивает значения из БД; local - сервис запрашивает данные единожды при загрузке и фильтрация происходит по загруженному набору значений.
queryparam	Наименование параметра при вводе значения в combobox для фильтрации	по какому ключ-значению из сервиса combobox производится фильтрация.

Пагинация значений combobox настраивается с помощью атрибута pagesize.

Поле "Изображение" (Field Image)

Поле для отображения изображений в модальном окне (Window), либо в панели (Panel).

Параметры

1. column - указывает на атрибут в объекте с данными.
Должен содержать ссылку на изображение в формате:

```
http://... https://...
```

либо изображение в формате base64:

```
data:image/jpeg;base64,...
```

("data:image/jpeg;base64," должно быть обязательно указано)

2. height - устанавливает высоту изображения (статическая высота в пикселях, px указывать не нужно))
3. width - устанавливает ширину контейнера для изображения (ширина поля. Целое число от 1% до 100%. Обязательно добавлять %)

Особенности и рекомендации

При наведении мышкой на изображение в нижней части отобразиться кнопка, которая позволяет скачать изображение.

По умолчанию изображение занимает всю возможную высоту без нарушения пропорций. В случаях, когда изображение по ширине уже контейнера, оно центрируется посередине. Если изображение меньше высоты контейнера оно будет отображаться оригинального размера, без растягивания. Не рекомендуется указывать и высоту и ширину одновременно - это может привести к искажению первоначальных пропорций.

По умолчанию имя файла "Изображение". Для того что-бы указать другое название файла, которое будет отображаться при скачивании, в объекте с данными должно быть поле в формате: {column}_filename, с названием в строчном формате. Например, для column=cv_image должно быть поле cv_image_filename.

Поле "Песочные часы" (Field Item Selector)

Поле для создания выборки данных.

Компонент используется для создание объекта типа "Песочные часы".

Настройте благоустройства

Справочник благоустройств

Тип благоустройства ↑

- Внутренняя система водоснабжения
- Наружная система водоснабжения
- Оборудован ваннами длиной 1500-1550 мм с душем
- Оборудован ваннами длиной 1650-1700 мм с душем
- Оборудован ваннами с душем
- Оборудован водонагревателями
- Оборудован водоразборной колонкой
- Оборудован газовыми водонагревателями
- Оборудован газовыми плитами
- Оборудован душами

...

Благоустройства на объекте обслуживания

Тип благоустройства ↑

- Закрытая система водоснабжения
- Индивидуальное отопление
- Оборудован ваннами без душа
- Оборудован ваннами сидячими длиной 1200 мм с душем

...

ЗАКРЫТЬ

Высота поля определяется атрибутом height.

Порядок настройки объекта типа "Песочные часы" можно посмотреть [тут](#)

Поле "Флаг" (Field Checkbox)

Поле для включения/выключения признака.

Если чек-бокс включен, то на бэк уходит значение "1", если выключен - "0".

Отображать в меню	<input checked="" type="checkbox"/>
Статическая страница	<input type="checkbox"/>

Поля с датами

Используется для отображения дат в формате ISO 8601.

Основные настройки полей [тут](#)

Настройки

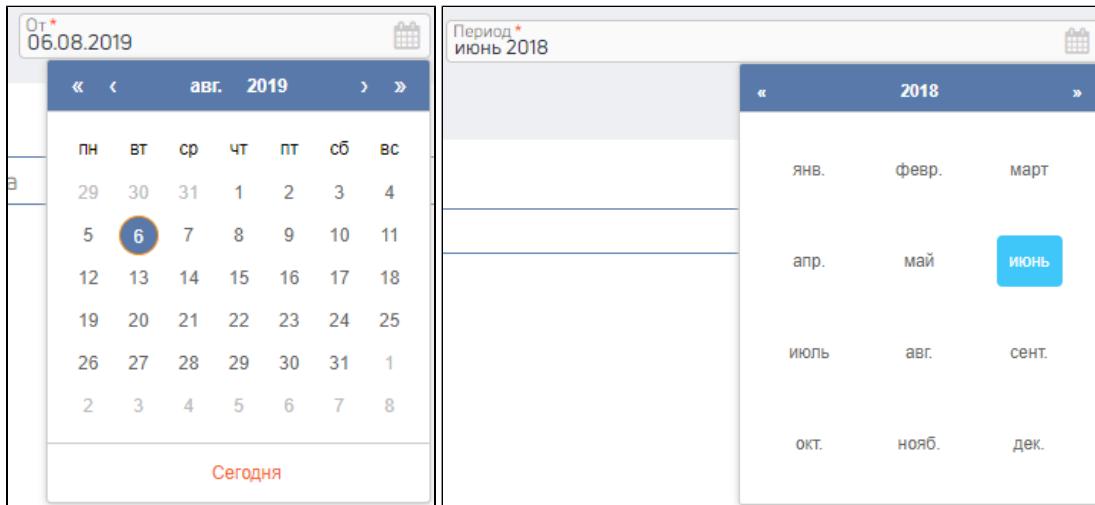
Формат даты задается с помощью атрибута format.

Возможные значения:

- 1 - ГГГГ;
- 2 - МММ ГГГГ;
- 3 - ДД.ММ.ГГГГ;
- 4 - ДД.ММ.ГГГГ ЧЧ:00;
- 5 - ДД.ММ.ГГГГ ЧЧ:МИ;
- 6 - ДД.ММ.ГГГГ ЧЧ:МИ:СС.

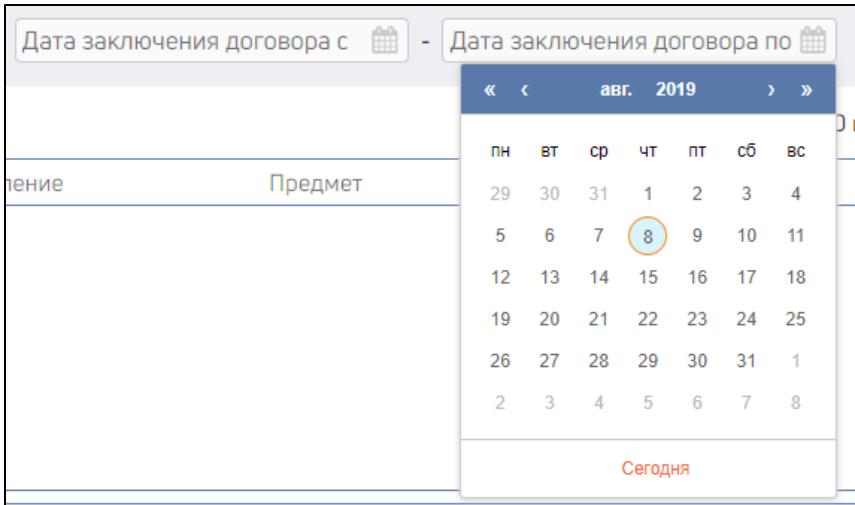
Поле "Дата" (Field Date)

Поле для отображения дат в формате ISO 8601.



Поле "Период" (Field Period Date)

Двойное поле для отображения периода дат в формате ISO 8601.



Основные настройки

Наименование атрибута	Комментарий
columnstart	Ключ-значение начала периода
columnend	Ключ-значение конца периода

Поведенческие настройки

Настройка behavior-атрибутов для блокировки/скрытия объекта класса Field Period Date влияет на отображение обоих полей (Дата начала и окончания периода) объекта.

Если необходимо настроить поведение одного из полей компонента (например, поле начала периода), необходимо:

1. создать в объекте класса Field Period Date дочерний объект класса Field Date;
2. указать для него значение атрибута column равное значению columnstart;
3. настроить необходимые behavior-атрибуты (setglobal, getglobal, etc.) для дочернего Field Date.

Пустое пространство (Empty Space)

Поле для резервирования пространства между полями. не отображается в интерфейсе.

С его помощью можно настраивать расположение полей в контейнерах (Box, Panel, Filter Panel).

Ширина Empty Space настраивается с помощью атрибута width.

Текстовые поля

Поля для работы с текстовыми данными.

- Поле "Многострочный текст" (Field Textarea)
- Поле "Пароль" (Field Password)
- Поле "Текст" (Field Text)
- Расширенный текстовый редактор (Field WYSIWYG)
- Скрытое поле (Field Hidden)

Основные настройки полей [тут](#)

Поле "Многострочный текст" (Field Textarea)

Текстовое поле для ввода и отображения нескольких строк текста.

Поле "Пароль" (Field Password)

Компонент для ввода и хранения паролей и других конфиденциальных данных.



Пароль учетной записи БД *

.....

eye icon



Пароль учетной записи БД *

5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8

X ⌂

Кнопка "глаз" позволяет скрывать/отображать данные поля.

Поле "Текст" (Field Text)

Поле для ввода/отображения текстовых данных.

Основные настройки полей [тут](#)

В атрибуте imask указывается маска ввода данных.

Например, (999) 999-99-99

Расширенный текстовый редактор (Field WYSIWYG)

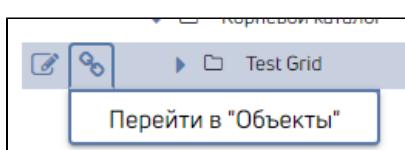
Скрытое поле (Field Hidden)

Текстовое поле, которое не отображается в интерфейсе.

Строка в меню "Переход на страницу" (Menu Redirect)

Компонент для включения в вызываемое меню при переходе на страницу.

Используется в паре с переиспользуемым объектом Menu Redirect Column Icon << Reusable.



columnsfilter - Список полей, по которым будет происходить фильтрация данных. Перечисление через запятую без пробела

redirecturl - URL страницы, на которую будет переход

Подробнее о реализации функции перехода на страницу [тут](#)

графики

contentview - влияет на расположение графика и легенды при первой загрузке страницы. hbox = hbox-wrap

contentwidth - влияет на ширину полей в легенде, если она сверху или снизу. Никак не влияет на ширину полей, если легенда слева или справа.

Работа с компонентами конструктора

- Redirect извне с подстановкой фильтров
- Redirect на страницу
- Reusable objects (Переиспользуемые объекты)
- Глобальные переменные

- Применение глобальных переменных
- Загрузка/выгрузка файлов по кнопке
- Как настроить объект "Песочные часы"

Redirect извне с подстановкой фильтров

Для интеграции с другими продуктами в платформе предусмотрен механизм перехода извне. Это позволяет переходить на страницу CORE, а также передавать значения фильтров.

Для перехода создана системная страница - redirect.

Для получения ссылки перехода необходимо подставить зашифрованный JSON с фильтрами в URL вида: *http://адрес core/redirect /зашифрованный JSON*.

Шифрование производится с помощью метода encodeURIComponent(btoa(unescape(encodeURIComponent(str))))), где str - строка, преобразованная через JSON.stringify.

Формат строки

1. page - адрес страницы на которую необходимо делать переход. Задается по принципу атрибута redirecturl
2. filter - задает передаваемые данные на страницу. Могут применяться как стандартные поля ввода (column) так и глобальные переменные.

Пример:

1. Исходный набор данных: {"page": "1", "filter": {"cv_name": "Account Grid"}}
2. Производим шифрование: encodeURIComponent(btoa(unescape(encodeURIComponent(JSON.stringify({"page": "1", "filter": {"cv_name": "Account Grid"}})))))
3. Полученная строка: "eyJwYWdlIjoiMSIsImZpbHRlcil6eyJjdI9uYW1lIjoiQWNjb3VudCBHcmklIn19"
4. Формируем ссылку для перехода: *http://адрес core/redirect /eyJwYWdlIjoiMSIsImZpbHRlcil6eyJjdI9uYW1lIjoiQWNjb3VudCBHcmklIn19*

Redirect на страницу

- Redirect на страницу по кнопке-колонке в гриде
 - Порядок настройки
- Redirect на страницу по гиперссылке
 - Порядок настройки
 - Принцип записи значений при переходе
 - Алгоритм поиска при Redirect

Redirect на страницу по кнопке-колонке в гриде

Данный функционал позволяет осуществлять переходы с одной страницы на другую с фильтрацией главного грида по

заданным параметрам при нажатии на кнопку-колонку .

Например, при переходе со страницы Страницы на страницу Объекты необходимо отобразить конкретный объект.

Для этого в фильтр на странице назначения передается id конкретного объекта (в данном примере - в [глобальную переменную](#), которую получает грид SYS Grid Object << DO NOT CHANGE через атрибут getglobal).

Порядок настройки

Для настройки переходов на страницы:

1. добавить в грид, из которого будет осуществляться переход, кнопку-колонку Menu Redirect Column Icon << Reusable.
2. добавить в тот же грид Объект класса Menu Redirect. для указанного объекта выполнить следующие настройки:
 - в атрибуте redirecturl указать URL страницы, на которую осуществляется переход по гиперссылке. Пример: bfl_1
 - в атрибуте columnsfilter указываем список полей, которые будут использоваться для фильтрации при переходе через запятую без пробела. Синтаксис аналогичен настройкам columnsfilter для гиперссылок

Если в выбранный грид нужно добавить еще один переход на другую страницу, то необходимо добавить в грид еще один объект класса Menu Redirect с соответствующими настройками.

Redirect на страницу по гиперссылке

Данный функционал позволяет осуществлять переход на страницу с заданными параметрами фильтрации при нажатии на гиперссылку.

Гиперссылки могут быть добавлены:

- в текстовые поля (Field Text);



- текстовые колонки (Column String).

Номер
1444
541
111
111
4343
10234-0

Порядок настройки

Для настройки гиперссылок необходимо:

1. на странице Страницы выбрать колонку либо поле, в котором будет содержаться гиперссылка
2. в атрибуте redirecturl указать URL страницы, на которую осуществляется переход по гиперссылке. Пример: bfl_1
3. в атрибуте columnsfilter указываем список полей, которые будут использоваться для фильтрации при переходе через запятую без пробела.

Пример:

ck_mo_ca=ck_mo_room,ck_mo_parent=ck_mo, где

ck_mo_ca, ck_mo_parent - поля страницы, с которой осуществляется переход по гиперссылке. Данные указанных полей будут присвоены полям ck_mo_room и ck_mo соответственно при переходе.

Важно!

В сервисе главного грида должна содержаться фильтрация по полям ck_mo_room и ck_mo (для страницы, на которую осуществляется переход).

Принцип записи значений при переходе

1. Собирается объект со значениями на переход (params)
2. Из params устанавливаются значения в поля при нахождении таких полей, при этом эти значения из params удаляются. Таким образом в params остаются только те значения, которые не должны явно влиять на значения полей.
3. Все оставшиеся параметры, после удаления, попадают в глобальные переменные. Таким образом оставляю только те переменные, которые должны влиять не явно на запросы (например getglobalstore)

Алгоритм поиска при Redirect

1. Очищаем все ранее примененные фильтры на странице, на которую осуществляется переход, независимо от того открыта она или закрыта.
2. Проставляем значения defaultvalue и defaultvaluequery в поля фильтра
3. Проставляем значения из columnsfilter в поля фильтра
4. Сеттим глобалки на полях фильтра, если необходимо
5. Геттим глобалки на полях фильтра, если необходимо
6. Осуществляем сам поиск.

Reusable objects (Переиспользуемые объекты)

Reusable object (переиспользуемый объект) - объект, который используется множество раз в разных местах системы путем привязки этого объекта напрямую к странице.

Основная идея состоит в том, чтобы не программировать и не рисовать типовые объекты несколько раз, вместе с этим сократив затраты на разработку и поддержку.

Reusable objects располагаются на верхнем уровне в иерархии объектов (т.е. не имеет родителя) и имеет пометку "<< Reusable".

Для того, чтобы подключить переиспользуемый объект к странице, необходимо:

1. выбрать родительский объект для Reusable-объекта на странице Страницы;
2. Нажать кнопку "Добавить" и в списке объектов выбрать нужный Reusable-объект;
3. Сохранить.

При настройке атрибутов Reusable-объекта следует однозначно понимать, что изменение на "Объекте" повлияет на все страницы, где используется этот Reusable; а переопределение атрибута на "Странице" - только на конкретную страницу. В первом случае изменения сохраняются в t_object_attr, во втором - в t_page_object_attr.

Глобальные переменные

Данный функционал расширяет возможности передачи данных из одного модуля метамодели в другой. Глобальные переменные объявляются только в разрезе одной Страницы. Поэтому настройка глобальных переменных производится на странице Страницы.

- Инициализация
- Именование глобальных переменных

Инициализация

Важно!

Перед инициализацией глобальной переменной необходимо указать ее наименование и описание на вкладке "Глобальные переменные на странице" на странице Страницы.

Имя глобальной переменной должно всегда начинаться с "g". Например, gk_id.

Именование глобальных переменных

За основу должен быть взять принцип именования переменных как в БД:

1. gv_value - текстовое значение
2. gd_date - текстовое представление даты
3. gn_number - числовое значение
4. gt_zona - текстовое представление даты с указанием временной зоны
5. gl_boolean - булевое значение, должно быть 0 или 1. Допускается пустоту.
6. gk_id - первичный ключ, может принимать значения как в виде строки, так и числовое

Применение глобальных переменных [тут](#)

Применение глобальных переменных

- Перед инициализацией
- Инициализация
- Атрибут Getglobal
 - Конкатенация ("склеивание") значений
- Атрибут Getglobalstore
- Поведенческие (Behavior) атрибуты

Перед инициализацией

Перед использованием глобальной переменной необходимо добавить ее в список Глобалок на странице. [Подробнее](#)

Инициализация

Для инициализации глобальной переменной используется атрибут **setglobal**.

- При объявлении setglobal на самом гриде глобальная переменная будет передавать значение ck_id выбранной записи в гриде.
- Если необходимо, чтобы в качестве глобальной передавалось значение, отличное от ck_id, setglobal объявляется в виде:

"cv_name=gv_name", где

cv_name - наименование колонки (column), значение которой будет передаваться в качестве глобальной переменной,

gv_name - наименование глобальной переменной.

- Установить setglobal можно и для самой колонки, если она есть в гриде. Тогда в setglobal данной колонки указывается только наименование глобальной переменной.
- Также возможно задать несколько глобальных переменных в setglobal для гридов, Combobox, Field Grid, etc. В этом случае глобальные переменные указываются через запятую без пробела:

cv_name=gv_name, cv_comment=gv_comment

Атрибут Getglobal

Если необходимо, чтобы значение глобальной переменной передавалось конкретному полю/колонке, указываем наименование глобальной переменной в атрибуте getglobal конкретного поля/колонки.

Конкатенация ("склеивание") значений

Использование глобальных переменных позволяет осуществлять конкатенацию значений нескольких полей в одно.

1. объявить глобальные переменные, используя атрибут setglobal
2. указать условия конкатенации в атрибуте getglobal нужного объекта (поля)

Пример:

Значение атрибута getglobal = gv_name || '555cnh', где

|| - символ склеивания,

'555cnh' - любой текст (указывается в одинарных кавычках),

gv_name - наименование глобальной переменной (указывается без кавычек).

Конкатенированное значение поля (допустим, gv_name = глобальная): глобальная555cnh

Атрибут Getglobaltostore

При указании наименования глобальной переменной в атрибуте getglobaltostore её значение будет передаваться в filter сервиса получающего объекта. Используется для объектов, вызывающих сервисы (например, Grid, Field Combobox, etc.)

Поведенческие (Behavior) атрибуты

Т.н. behavior-атрибуты влияют на поведение объекта на странице.

Наименование атрибута	Комментарий

disabledrules	Условие, при выполнении которого объект станет неактивным (недоступным для редактирования). Используется для полей/кнопок/колонок.
hiddenrules	Условие, при выполнении которого объект будет скрыт для отображения.
readonlyrules	Условие, при выполнении которого объект будет доступен только для чтения. Функции редактирования будут недоступны.
requiredrules	Условие, при выполнении которого объект станет обязательным для заполнения
reqcountrules	Условие, при выполнении которого изменяется количество обязательных для заполнения полей в Field Group. Имеет приоритет над reqcount.

Синтаксис:

'==' - равно
 '!= - не равно
 '>' - больше
 '<' - меньше
 '&&' - и
 '| |' - или

'!gk_mo' - пустое поле, на котором установлена глобальная переменная

'ck_id=="setting" | ck_id=="remove"' - при сравнение со строкой нужно оборачивать только в двойные кавычки, одинарные использовать нельзя

('debt' in gk_d_anomaly)- когда необходимо сравнить значение с массивом.

Пример:

Значение атрибута hiddenrules = gk_mo>1

Поведение: Если глобальная переменная gk_mo больше 1, то поле скроется и не будет отображено в интерфейсе.

Окно для настройки поведенческих атрибутов позволяет сразу проводить валидацию введенного значения:

Редактирование поведенческих атрибутов

Выполнение

Результат
false

Переменные

Тип данных в результате: boolean

СОХРАНИТЬ
ОТМЕНА

Редактирование поведенческих атрибутов

Выполнение

Исходный код
gv_name==1

Результат
true

Тип данных в результате: boolean

Переменные

Значение для gv_name
1

СОХРАНИТЬ

ОТМЕНА

Загрузка/выгрузка файлов по кнопке

Для реализации импорта/экспорта файлов в/из Core необходимо:

1. создать кнопку (Button), которая будет вызывать функцию импорта файла;
2. Настраиваем ее атрибуты:

Наименование атрибута	Возможные значения	Комментарий
extraplugingate		Имя плагина шлюза, осуществляющего загрузку/выгрузку
filemode	single - одиночная загрузка; multi - возможность загрузки нескольких файлов сразу	Тип загрузки файла. Если не заполнен, то используется single
filetypes	doc,docx,pdf,zip,txt,ods,odt,xls,xlsx	Доступные типы файлов
maxfile		Максимальный размер файла в байтах
mode	7 - экспорт файлов из Core 8 - загрузка файлов в Core	Тип операции
updatequery	Обязательный атрибут	Имя сервиса, осуществляющего загрузку/выгрузку

3. место хранения файла определяется соответствующим плагином. Берется либо плагин, указанный в extraplugingate, либо плагин, который использует провайдер сервиса, указанного в updatequery

Пример сервиса:

```
select pkg_json_doc.f_modify_doc(pn_user => :sess_ck_id,
                                  pv_session => :sess_session,
                                  pc_json => :json,
                                  pb_file => :upload_file,
                                  pv_filename => :upload_file_name)
as result
```

Как настроить объект "Песочные часы"

Объект типа "Песочные часы" можно создать в родительском объекте класса:

1. Окно (Window) - для кастомных окон;
 2. Колонка "Строка" (Column String) - для автособираемых окон;
 3. Панель (Panel).
-

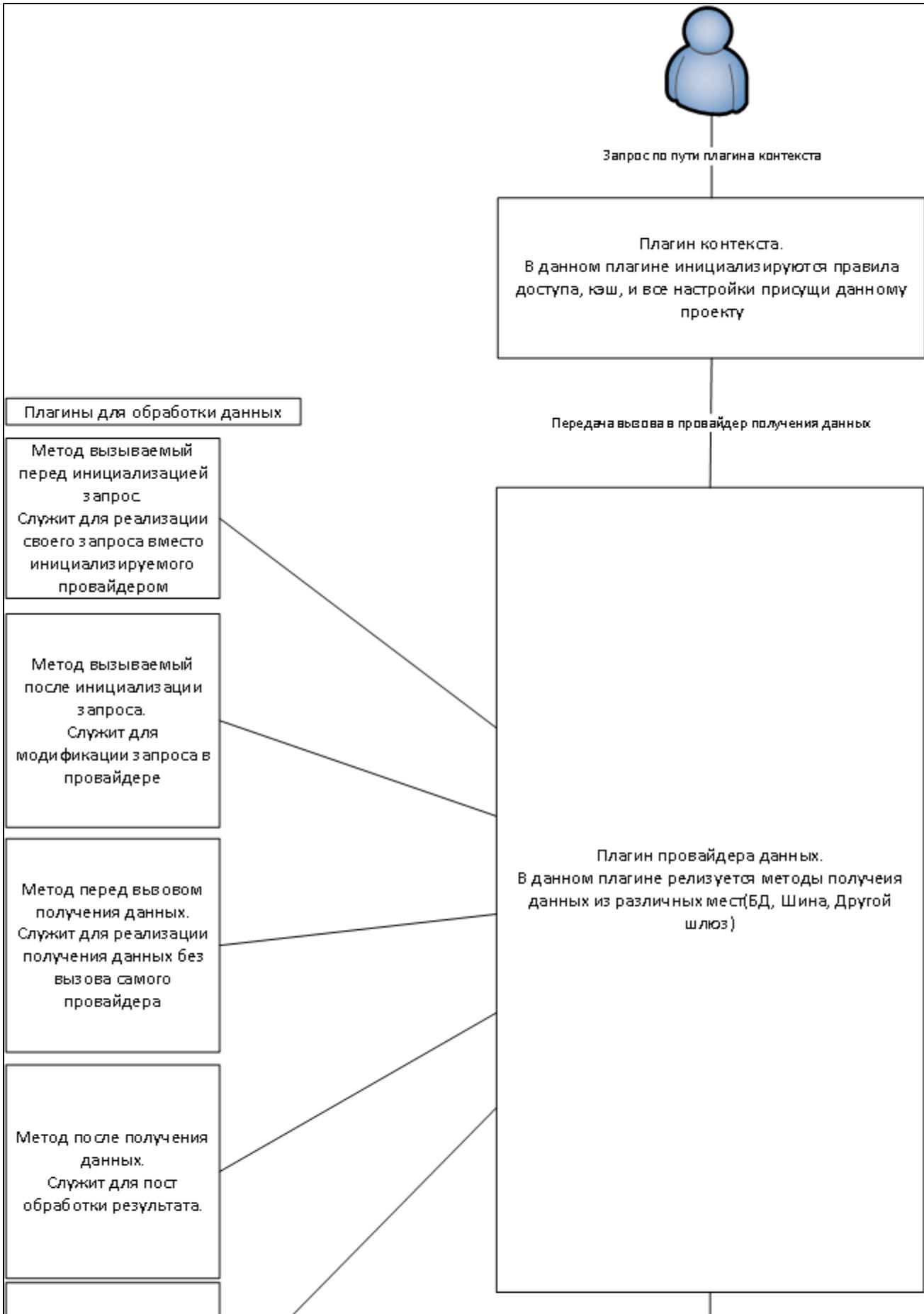
Для создания Объекта "Песочные часы" необходимо:

1. Создать объект класса Поле "Песочные часы" (Field Item Selector);
2. В Field Selector создаются 2 грида:
 1. доступные для выбора данные;
 2. выбранные данные.
3. Настройки для гридов:
 1. Уникальный идентификатор (ck_id) записи в обоих гридах должен описывать одну и ту же сущность;
 2. модификатор сервиса должен уметь работать с массивом данных;
 3. указываем значение атрибута selmode = MULTI на обоих гридах;
 4. чтобы спрятать управляющие кнопки на гридах, указываем значение атрибута hideactions = true на обоих гридах.

Шлюз

Каталог для описания настроек шлюза и среднего слоя

Описание работы шлюза





Последовательность работы.

1. Запрос от пользователя приходит на плагин контекста.
2. В плагине контекста проверяются доступы, кэшируются страницы и инициализируются данные для последующей передачи вызова в провайдер данных, либо получаем сразу результат для ответа. Контекст выбирает, какому провайдеру передать запрос на выполнение.
3. В плагине провайдера осуществляется инициализация запроса, вызов запроса с получением данных. Т.е. провайдер работает прямолинейно: получает запрос, передает его в БД/шину/другой шлюз, получает ответ и передает его обратно в контекст.
4. При необходимости провайдер вызывает плагин/плагины. Плагины данных работают на этапе работы провайдера. Если провайдеру для обработки необходимо как-то модифицировать запрос/ответ, то дополнительно привлекается плагин. Например, при печати отчетов Провайдер вызывает соответствующие плагины uspo для распаковки json, передачи его в jasper, преобразования ответа от jasper.
5. Данные отдаются пользователю.

Есть возможность реализации методов на любом этапе вызова.

Подробнее о каждом плагине можно почитать [здесь](#)

Плагины шлюза

Создание

Плагины создаются, реализуя абстрактных классов:

NullContext - абстрактный класс плагинов контекста
 NullPlugin - абстрактный класс плагинов работы с данными
 NullProvider - абстрактный класс плагинов провайдеров данных
 NullAuthProvider - абстрактный класс плагинов авторизационных провайдеров
 NullScheduler - абстрактный класс плагинов планировщика
 NullEvent - абстрактный класс автономных служб

Плагины контекста

CoreOracleContext

Плагин контекста CORE.

В данном плагине кэшируются доступы, страницы, ошибки, запросы. Осуществляется проверка доступа. Отдача готовой страницы из кэша. Основные настройки относятся к БД.

Настройки

Наименование параметра	Описание
user	Наименование учетной записи БД
password	Пароль учетной записи БД

- Создание
- Плагины контекста
 - CoreOracleContext
 - CoreOracleIntegrationContext
- Плагины провайдеров авторизации
 - AuthCrmWs
 - CoreAuthPG
 - PKOAuth
 - AdAuth
- Плагины провайдеров
 - AdminGate
 - CoreOracleIntegration
 - OracleDb
 - PostgreSQLDb
- Плагин работы с данными
 - JsonRowColumnExtractor
 - PrepareQueryPlugin
 - USPOIntegration

connectString	Строка подключения к БД
poolMin	Минимальное количество коннектов к БД в пуле
poolMax	Максимальное количество коннектов к БД в пуле

CoreOracleIntegrationContext

Плагин инициализации интеграция с БФЛ внешними сервисами.

В данном плагине кэшируются доступы и запросы.

▼ Настройки

Наименование параметра	Описание
user	Наименование учетной записи БД
password	Пароль учетной записи БД
connectString	Строка подключения к БД
poolMin	Минимальное количество коннектов к БД в пуле
poolMax	Максимальное количество коннектов к БД в пуле

Плагины провайдеров авторизации

AuthCrmWs

Плагин авторизации СУБК через шину.

▼ Настройки

Наименование параметра	Описание
queryAuth	Наименование запроса проверки авторизации
queryMetaUsers	Наименование запроса получения мета информации
queryUsersActions	Наименование запроса получения всех экшенов
queryUsersDepartments	Наименование запроса получения департаментов по юзеру
queryDepartments	Наименование запроса получения всех департаментов
queryToken	Наименование запроса получения token
queryGetCrmUrl	Имя запроса получения ссылки на сувк
cronReset	Время когда сбрасывается кэш. Порядок ввода Linux: * * * * *
cnSystem	Ид системы
clAudit	Признак аудита
sessionDuration	Время жизни сессии
urlCrmTemplate	Шаблон ссылки

AuthMock

Плагин для реализации заглушки авторизации для новых проектов.

▼ Настройки

Наименование параметра	Описание
adminPassword	Пароль администратора
adminUser	Наименование учетки администратора
viewUser	Наименование учетки просмотра

ProjectTiiAuth

Плагин авторизации в проектном учете.

▼ Настройки

Наименование параметра	Описание
user	Наименование учетной записи БД
password	Пароль учетной записи БД
connectString	Строка подключения к БД
poolMin	Минимальное количество коннектов к БД в пуле
poolMax	Максимальное количество коннектов к БД в пуле

CoreAuthOracle

Плагин авторизации структура Core

▼ Настройки

Наименование параметра	Описание
user	Наименование учетной записи БД
password	Пароль учетной записи БД
connectString	Строка подключения к БД
poolMin	Минимальное количество коннектов к БД в пуле
poolMax	Максимальное количество коннектов к БД в пуле

CoreAuthPG

Плагин авторизации структура Core

▼ Настройки

Наименование параметра	Описание
------------------------	----------

connectString	Строка подключения к БД
poolMax	Максимальное количество коннектов к БД в пуле

PKOAuth

Плагин авторизации в ПКО представляет связку CoreAuth и AdAuth

Настройки

Наименование параметра	Описание
user	Наименование учетной записи БД
password	Пароль учетной записи БД
connectString	Строка подключения к БД
poolMin	Минимальное количество коннектов к БД в пуле
poolMax	Максимальное количество коннектов к БД в пуле

Наименование параметра	Описание
adBaseDN	Начальный уровень поиска в ldap. Пример: dc=example,dc=com
adDefaultAction	Список экшенов по умолчанию. Пример: 150,250,300
adLogin	Логин УЗ доступа к ldap
adMapGroups	Мапинг групп и экшенов пользователя. Пример: NameGroup=900,200;NameGroup=100,215
adMapUserAttr	Мапинг атрибутов пользователя. Пример: cv_login=sAMAccountName;cv_name=cn;cv_surname=sn;cv_email=mail;cv_cert=userCertificate
adPassword	Пароль УЗ доступа к ldap
adUrl	Урл к ldap. Пример: Пример: ldap://dc.domain.com

AdAuth

Плагин авторизации в ActiveDirectory

Настройки

Наименование параметра	Описание
adBaseDN	Начальный уровень поиска в ldap. Пример: dc=example,dc=com
adDefaultAction	Список экшенов по умолчанию. Пример: 150,250,300
adLogin	Логин УЗ доступа к ldap
adMapGroups	Мапинг групп и экшенов пользователя. Пример:

	NameGroup=900,200;NameGroup=100, 215
adMapUserAttr	Мапинг атрибутов пользователя. Пример: cv_login=sAMAccountName; cv_name=cn;cv_surname=sn; cv_email=mail;cv_cert=userCertificate
adPassword	Пароль УЗ доступа к ldap
adUrl	Урл к ldap. Пример: Пример: ldap://dc.domain.com

Плагины провайдеров

AdminGate

Плагин администрирования шлюза.

▼ Настройки

Наименование параметра	Описание
cv_riak_url	Ссылка на хранилище riak
cct_buckets	Настройки хранилища в формате json

CoreOracleIntegration

Плагин работы интеграции БФЛ с внешними системами.

▼ Настройки

Наименование параметра	Описание
timeout	Время ожидания внешнего сервиса
needSession	Необходимость проверки авторизации для запросов
proxy	Прокси сервер для вызова внешних систем (если необходимо)
user	Наименование учетной записи БД
password	Пароль учетной записи БД
connectString	Строка подключения к БД
poolMin	Минимальное количество коннектов к БД в пуле
poolMax	Максимальное количество коннектов к БД в пуле

OracleDb

Плагин вызова запросов в БД Oracle.

▼ Настройки

Наименование параметра	Описание

core	режим с инициализацией темповых таблиц
user	Наименование учетной записи БД
password	Пароль учетной записи БД
connectString	Строка подключения к БД
poolMin	Минимальное количество коннектов к БД в пуле
poolMax	Максимальное количество коннектов к БД в пуле

PostgreSQLDb

Плагин вызова запросов в БД Postgres.

▼ Настройки

Наименование параметра	Описание
core	Инициализация по схеме CORE
connectString	Строка подключения к БД
poolMax	Максимальное количество коннектов к БД в пуле

Плагин работы с данными

JsonRowColumnExtractor

Распаковываем ответ из определенных колонок из строки в json.

▼ Настройки

Применяется одна из:

Наименование параметра	Описание
columns	Наименования колонок, которые надо распаковать (через запятую)
extractSingleColumn	Признак, что надо распаковать колонку (если она одна, является строкой и начинается с { или [)

PrepareQueryPlugin

Модификация запроса, применение фильтров.

USPOIntegration

Плагин печати через УСПО.

▼ Настройки

Наименование параметра	Описание
timeout	Время ожидания
urlReceiver	URL JS

<code>prefixPath</code>	Префикс пути для формирование ссылки на скачивание
<code>queryNotification</code>	Наименование запроса для оповещения
<code>maxExcelRows</code>	Максимальное количество строк при печати в Excel
<code>exportUrl</code>	UUID на отчет excel

Разработчику

Репозиторий

<https://github.com/essence-community/core-backend>

- Репозиторий
- Структура репозитория
- Подготовка к работе
- Сборка приложения

Структура репозитория

`source`

- cert Сертификаты для построения защенного канала кластера
- contexts Плагины контекста
- events Плагины служб оповещений и автономных служб
- libs Библиотеки
- plugininf Пакет общих классов и интерфейсов плагинов
- plugins Плагины обработки и модификации данных
- providers Плагины получения данных
- resources Настройки
- schedulers Плагины планировщика
- server Реализация сервера
 - admin реализация процесса администрирования кластера
 - core общие классы
 - events реализация процесса автономных служб
 - http реализация кластера http сервера
 - localDbNode реализация процесса локально NeDb
 - master реализация основного управляющего процесса
 - schedulers реализация процесса планировщика

Подготовка к работе

Для разработки надо установить Node 10+ и yarn 1.9+.

Для работы с oracle необходимо установить клиент(битность аналогичной nodejs) не ниже 12.2 и прописать его в PATH.(Подойдет Instant Client).

Сборка всех зависимостей вызывать из папки source:

```
yarn install
```

Сборка приложения

Перейти в папку source и вызвать:

```
yarn build
```

Соберет в папку bin

Система модулей

- Зачем нужны модули
- Как добавить свой модуль
- Дополнительные возможности
- Что такое метамодель
- Что такое глобальные переменные
- Какие типы данных доступны
- Параметры который приходят в модуль
- Как использовать другие модули в своих модулях

Зачем нужны модули

Модульная система предназначена для расширения существующего набора компонентов посредством добавления новых пакетов (модулей) через систему администрирования либо изменения существующих классов.

Как добавить свой модуль

Для добавления модуля необходимо:

1. Реализовать модуль как отдельный пакет;
2. Использовать `setComponet` для установки компонента как модуля;
3. ... в стадии разработки.

Дополнительные возможности

Ядро конструктора использует react с динамическим обновлением данных и возможностью изменения темы

1. React - служит для отображения данных и манипуляции с DOM элементами. Пакет должен отдавать react компонент в качестве модуля;
2. Mobx - утилита для управления состоянием приложения в реальном времени. С помощью этой утилиты можно реагировать на изменения окружающих модулей и распространять данных из своего модуля;
3. StoreBaseModel - базовая модель для создания хранилища по обмену данными между серверов и клиентом. Так же служить для реализация связей по средствам мастеров. Добавляет автоматическое обновление данных при изменении мастера.
4. Material-ui - утилита для визуализации простых компонентов и работы с css. На базе утилиты построена система изменения темы и цветовой палитры приложения.
5. PageModel (PageStore) - хранилище состояния страницы, на которой находится модуль. Предоставляет доступ к другим модулям через хранилища (store) других модулей. Дополнительно хранит в себе открытые окна и состоянием видимости страницы.
6. ApplicationModel (ApplicationStore) - общее хранилище приложения. Служит для работы с модулем авторизации и уведомлением. Хранит информацию о всех активных страницах в приложении. Позволяет осуществлять переход между страницами с предоставляем информацией о предыдущей странице с помощью глобальных переменных.

Что такое метамодель

Метамодель приложения разделена на 4 базовых типа:

1. Атрибуты - описывают типы данных в настройках одного класса. Каждый атрибут может быть опциональным. При работе с атрибутом в модуле необходимо проверить наличие в нем данных, чтобы не вызвать критическую ошибку.
2. Классы - описывают состояние модуля. Класс может включать в себя все атрибуты или часть из них. Обязательным атрибутом является только type по которому происходит определение, к какому модулю относится данный класс.
3. Объекты - служат для объединения классов в один объект (например: таблица может включать в себя колонки для построения таблицы с нужными колонками для отображения). Вложенность одних объектов в другие определяется на уровне создания класса. Для каждого класса разрешается переопределить его атрибуты.
4. Страница - служит для построения страницы (PageStore) из объектов. Для каждого класса разрешается переопределить его атрибуты, при этом переопределение будет доступно только для выбранной страницы. Один и тот же объект может быть добавлен на несколько страниц сразу.

Что такое глобальные переменные

Для общения между модулями реализовано общее хранилище переменных (globalValues в PageStore).

Хранилище позволяет:

1. Записывать данные из любого модуля (при использовании одного и того же ключа в разных модулях согласованность данных не гарантирована, вся ответственность ложится на создателя метамодели).
2. Читать данные - при этом нужно понимать, что тип данных может быть любой: undefined, null, object, string, number, boolean, "array". При чтении данных нужно использовать reaction или autorun из mobx либо использовать mobx-react для подписывания render функции самого компонента на изменение данных.
3. Другие действия с данными - происходит посредством вызовов методов ObservableMap.

Глобальные переменные так же могут изменится при переходе между страницами посредством applicationStore

Какие типы данных доступны

Для работы со статической типизацией и утилитами предоставляется пакет constructor-share который включает:

1. Типы данных для атрибутов BuilderConfigType. Данная структура синхронизируется с базой данных и позволяет поддерживать код в актуальном состоянии.
2. ApplicationModelType - тип данных для работы с приложением и авторизацией. Доступно через pageStore модуля.
3. PageModelType - хранит состояние страницы.

Параметры который приходят в модуль

1. bc - BuilderConfigType хранит состояния класса переданного из метамодели;
2. pageStore - состояние страница, описано в типе PageModelType
3. hidden - (boolean) - состояние скрытия контента с помощью правил метамодели
4. disabled - (boolean) - состояние блокировки с помощью правил метамодели
5. readOnly - (boolean) - состояние "только для чтения" с помощью правил метамодели
6. visible - (boolean) - состояние отображения контента на странице. Если страница полностью не показывается, то visible будет false. Иначе передается по цепочке.

Параметры hidden, disabled, readOnly, visible могут быть изменены в зависимости от состояний промежуточных компонентов.

Как использовать другие модули в своих модулях

Для отображения дочерних компонентов использовать mapComponents при этом в каждый компонент необходимо обязательно передать pageStore и bc для правильной работы другого модуля. Так же нужно прокинуть по цепочке disabled, hidden, readOnly, visible при этом значения можно менять в зависимости от необходимости либо оставить какие пришли для совместимости.