

1. Back-end	3
1.1 Обработка ошибок	3
1.2 Сервис для Поля ввода Mask	5
1.3 Стандарт разработки	5
2. Front-end	19
2.1 Конструктор	19
2.1.1 Атрибут	19
2.1.2 Класс	22
2.1.3 Объект	26
2.1.4 Страница	31
2.2 Интерфейс и элементы управления	39
2.2.1 Авторизация	39
2.2.2 Глобальная панель	40
2.2.2.1 Основное меню, панель закладок (быстрого доступа)	41
2.2.2.2 Панель оповещения	42
2.2.2.3 Профиль пользователя	42
2.2.2.4 Информация о программе	43
2.2.3 Рабочая область	43
2.2.4 Элементы управления и их обозначение	43
2.2.5 Элементы ввода данных	50
2.2.6 Окна для ввода данных	52
2.2.7 Мастер ввода данных (песочные часы)	53
2.3 Компоненты конструктора	54
2.3.1 Гриды	55
2.3.1.1 Простой грид (Grid)	56
2.3.1.2 Грид типа "Дерево" (Grid Tree)	58
2.3.2 Кнопки (Button)	58
2.3.2.1 Собиратель кнопок (Button Collector)	61
2.3.3 Колонки	61
2.3.3.1 Колонка "Boolean" (Column Boolean)	62
2.3.3.2 Колонка "Дата" (Column Date)	63
2.3.3.3 Колонки-кнопки	63
2.3.3.3.1 Колонка "Флаг" (Column Checkbox)	63
2.3.3.3.2 Колонка-кнопка "Редактирование" (Column Edit)	64
2.3.3.3.3 Колонка-кнопка "Удаление" (Column Delete)	64
2.3.3.3.4 Колонка-кнопка для отображения значка детализации (Column Detail)	65
2.3.3.3.5 Колонка-кнопка с иконкой (Column Action Icon)	65
2.3.3.4 Текстовые колонки	66
2.3.3.4.1 Колонка "Дерево" (Column Tree)	66
2.3.3.4.2 Колонка "Строка" (Column String)	66
2.3.3.5 Числовые колонки	67
2.3.3.5.1 Колонка "Целое число" (Column Integer)	67
2.3.3.5.2 Колонка "Число" (Column Numeric)	67
2.3.4 Окно (Window)	68
2.3.4.1 Автособираемое окно	68
2.3.4.2 Кастомное окно	71
2.3.5 Панели	73
2.3.5.1 Динамическая панель (Dynamic Panel)	73
2.3.5.2 Контейнер типа "Box"	78
2.3.5.3 Контейнер типа «Панель» (Panel)	78
2.3.5.4 Панель детализации в гриде (Panel Detail)	79
2.3.5.5 Панель загрузки и отображения файлов (Panel File)	80
2.3.5.6 Панель с историчными данными (Panel History)	80
2.3.5.7 Панель со вкладками (Tab Panel)	83
2.3.5.8 Панель с фильтрами (Filter Panel)	85
2.3.5.9 Панель с шагами (Panel Road Map)	86
2.3.6 Поля	87
2.3.6.1 Встроенный фрейм (IFrame)	88
2.3.6.2 Групповые поля	88
2.3.6.2.1 Field Group	88
2.3.6.2.2 Field Set	89
2.3.6.3 Поле "Выпадающая таблица" (Field Grid)	89
2.3.6.4 Поле "Выпадающая таблица-дерево" (Field Grid Tree)	91
2.3.6.5 Поле "Выпадающий список" (Field Combobox)	92
2.3.6.6 Поле "Изображение" (Field Image)	94
2.3.6.7 Поле "Песочные часы" (Field Item Selector)	95

2.3.6.8 Поле "Флаг" (Field Checkbox)	95
2.3.6.9 Поля с датами	95
2.3.6.9.1 Поле "Дата" (Field Date)	96
2.3.6.9.2 Поле "Период" (Field Period Date)	96
2.3.6.10 Пустое пространство (Empty Space)	97
2.3.6.11 Текстовые поля	97
2.3.6.11.1 Поле "Многострочный текст" (Field Textarea)	97
2.3.6.11.2 Поле "Пароль" (Field Password)	98
2.3.6.11.3 Поле "Текст" (Field Text)	98
2.3.6.11.4 Расширенный текстовый редактор (Field WYSIWYG)	98
2.3.6.11.5 Скрытое поле (Field Hidden)	98
2.3.7 Стока в меню "Переход на страницу" (Menu Redirect)	98
2.3.8 графики	99
2.4 Компоненты базового представления	99
2.4.1 Приложение (Application)	99
2.4.2 Контейнеры для контента страниц (Pages и Pager)	100
2.4.3 Навигационная панель страницы (App Bar Panel)	101
2.4.4 Группы кнопок (Button Group)	101
2.4.5 Вкладки открытых страниц (Open Page Tabs)	102
2.4.6 Окно "Выдвижная панель" (Window Drawer)	102
2.5 Работа с компонентами конструктора	103
2.5.1 Redirect извне по сессии/токену	103
2.5.2 Redirect извне с подстановкой фильтров	104
2.5.3 Redirect на страницу	105
2.5.4 Reusable objects (Переиспользуемые объекты)	107
2.5.5 Глобальные переменные	108
2.5.5.1 Применение глобальных переменных	109
2.5.6 Загрузка/выгрузка файлов по кнопке	112
2.5.6.1 Построчный разбор при загрузке файлов XLSX и CSV	113
2.5.7 Как настроить объект "Песочные часы"	115
3. Шлюз	117
3.1 Описание работы шлюза	117
3.2 Плагины шлюза	120
3.3 Разработчику	126
3.4 Система оповещения	127
4. Система модулей	128
4.1 Модуль УСПО	130
5. Быстрый старт	141
5.1 Шаг 0. Создание и настройка провайдера	141
5.2 Шаг 1. Грид на отображение	144
5.3 Шаг 2. Добавление/редактирование/удаление данных в гриде	149
6. Внутренняя авторизация	153
6.1 Настройка действий	154
6.2 Настройка ролей	156
6.3 Настройка пользователей	161
6.3.1 Справочник дополнительных данных пользователей	165
6.4 Сброс провайдера авторизации	171
7. Локализация	172
7.1 Словарь	173
7.2 Список языков	177
8. Патчинг	179
8.1 Патчинг авторизации	179
8.2 Патчинг метаданных	182

Back-end

Эта страница описывает правила разработки и кодирования

Важно!

Front-end определяет уникальность сущности по параметру ck_id, поэтому любой запрос должен содержать уникальный идентификатор сущности (ck_id).

Важно!

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

- 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 ссылаемся 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 (незафиксированные в проектном документе) запрещены.

В наименованиях таблицы запрещено использовать множественное число (необходимо подобрать синоним или называть 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:

c<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 пользовательского 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 от_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
- Front-end определяет уникальность сущности по параметру ck_id, поэтому любой запрос должен содержать уникальный идентификатор сущности (ck_id)
- 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 l=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-полей (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
(
  ...
) 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 2x 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.

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

- Атрибут
- Класс
- Объект
- Страница

Структура метамодели

Основные компоненты:

1. Класс;
2. Объект;
3. Страница.



Атрибут

Атрибут – свойство **класса** либо **объекта**.

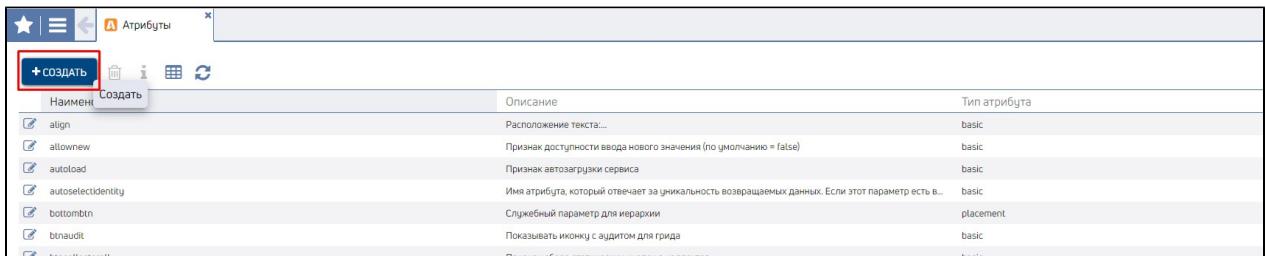
Каждому классу соответствует набор атрибутов.

- Создание атрибута
- Типы атрибутов
- Редактирование атрибута
- Удаление атрибута
- В каких классах используется

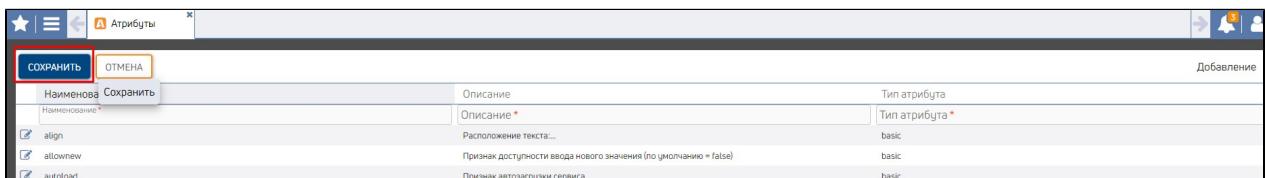
Создание атрибута

Для создания атрибута необходимо:

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



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



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:

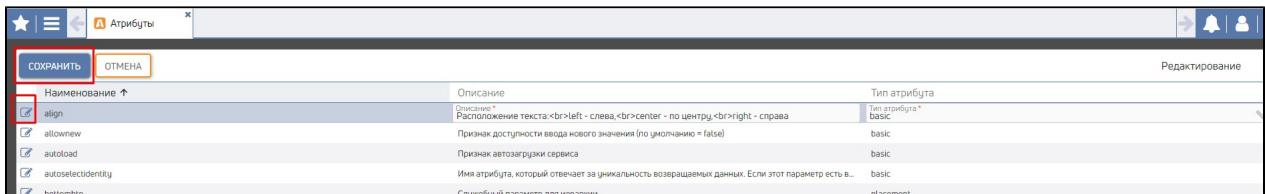
Наименование	Описание
basic	обычный атрибут
behavior	поведенческий атрибут для зависимости отображения от данных. Переопределяется только на страницах.
system	системный атрибут, не переопределяется для объекта/страницы
placement	место для размещения дочернего элемента
view	атрибут, влияющий на визуализацию

Редактирование атрибута

Для редактирования атрибута необходимо:

1. выбрать атрибут;

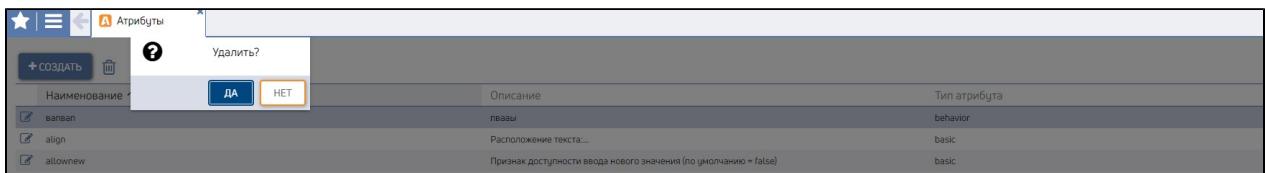
2. нажать на кнопку-колонку «Редактировать значение»;
3. внести необходимые изменения;
4. нажать кнопку «Сохранить».



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.

В каких классах используется

Для Просмотра классов, в которых используется тот или иной атрибут необходимо:

1. перейти на страницу «Атрибуты» (Меню -> CORE -> Конструктор GUI -> Классы);
2. выбрать атрибут, для которого необходимо уточнить список использующих классов;
3. в гриде "В каких классах используется" будет отображен список классов, использующих выделенный атрибут.

Наименование ↑			Описание	Тип атрибута
btprint			Отображение кнопки "Печатать" (true/false)	basic
btrefresh			Отображение кнопки "Обновить" (true/false)	basic
btnsettings			Отображение кнопки "Настройки" (true/false)	basic
checkboxmore			Признак "Добавить еще". Если не заполнен = false	behavior
childs			Служебный параметр для иерархии	placement
childwindow			Служебный параметр для иерархии	placement
ckwindow			Идентификатор для поиска окна в мастере	basic
collapsed			Свернута ли панель при инициализации ...	view
collapsible			Признак возможности сворачивания панели true/false	behavior
collectionvalues			Тип сбора значений...	basic
column			Наименование параметра для отображения данных и передачи в Modify	basic
columnend			Наименование параметра - конца периода	basic
columns			Служебный параметр для иерархии	placement

В каких классах используется	
Наименование класса ↑	Описание класса
Button	Кнопка
Column Action Icon	Колонка-кнопка с иконкой
Column Edit	Колонка-кнопка "Редактирование"
Window	Окно
Window Drawer	Окно для открытия слева / справа в виде drawer

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:

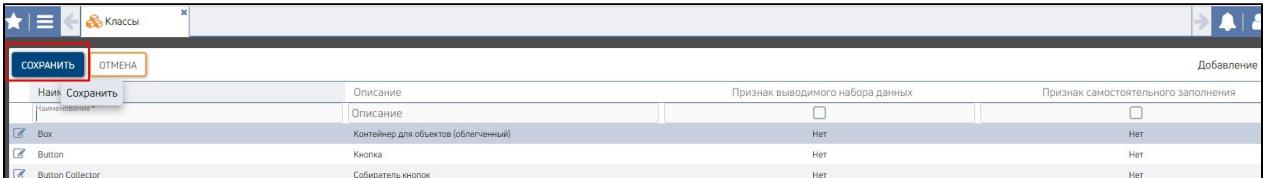
1. перейти на страницу «Классы» (Меню -> CORE -> Конструктор GUI -> Классы);
2. нажать кнопку «Создать»;



Нимено	Описание	Признак выводимого набора данных	Признак самостоятельного зап
Box	Контейнер для объектов (облегченный)	Нет	Нет
Button	Кнопка	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/скрытия деталей	Нет	Нет

3. заполнить поля;

4. нажать кнопку «Сохранить».



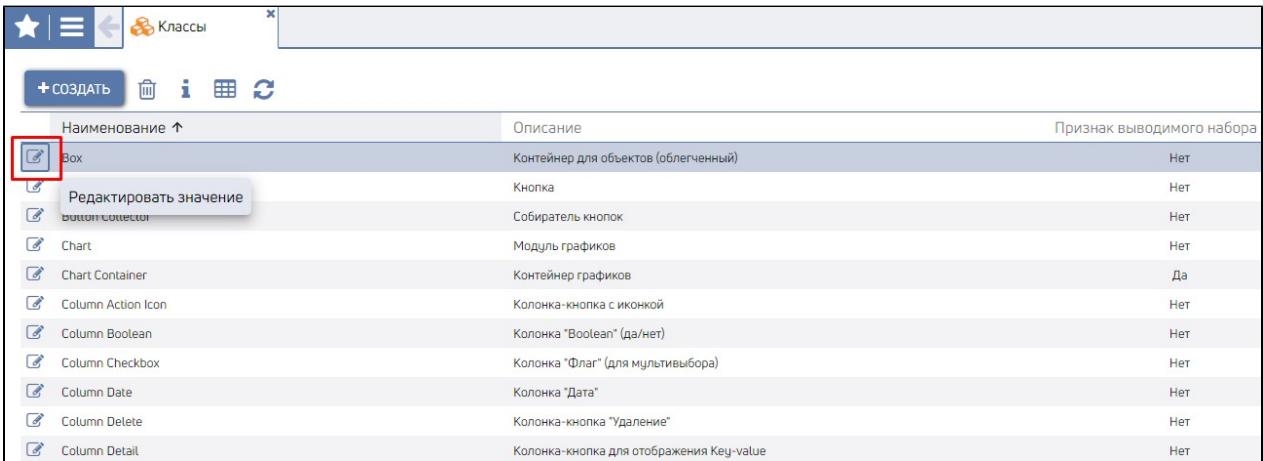
Доступные поля для заполнения:

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:

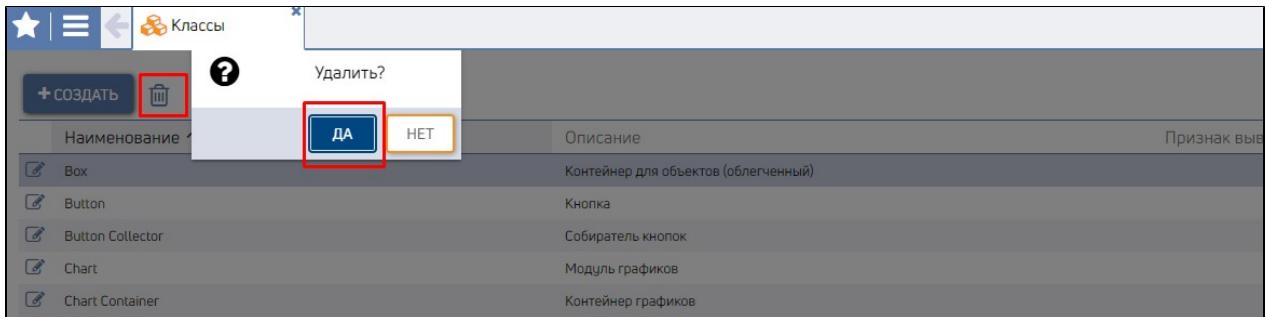
1. выбрать класс;
2. нажать кнопку-колонку «Редактировать значение»;
3. внести изменения;
4. нажать кнопку «Сохранить».



Удаление класса

Для удаления класса необходимо:

1. выбрать класс;
2. нажать кнопку «Удалить»;
3. 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:

1. выделить класс, для которого будут создаваться атрибуты;
2. выбрать вкладку «Атрибуты»;
3. нажать кнопку «Добавить»;

Наименование	Значение	Признак обязательности	Описание	Тип
align	right	Нет	Расположение текста...	view
column		Нет	Наименование параметра для отображения данных и пер...	basic
datatype	integer	Да	Тип данных колонки	system

4. заполнить поля;

5. нажать кнопку «Сохранить».

Добавление				
Атрибуты		Дочерние классы		Родительские классы
<input type="button" value="СОХРАНИТЬ"/> <input type="button" value="ОТМЕНА"/>				
Найменование	Значение	Признак обязательности	Описание	Тип атрибута
Найменование	true	<input type="checkbox"/>	Признак автозагрузки сервиса	basic
autoload			Служебный параметр для иерархии	placement
child				
placement				

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:

1. выделить класс, к которому будут добавляться дочерние классы;
2. выбрать вкладку «Дочерние классы»;
3. нажать кнопку «Добавить»;

Добавление				
Атрибуты		Дочерние классы		Родительские классы
<input type="button" value="+ Добавить"/> <input type="button" value="Удалить"/> <input type="button" value="Сохранить"/> <input type="button" value="Отмена"/>				
Наименование	Добавить та ↑	Наименование класса	Описание класса	
editors		Field Checkbox	Поле "Флаг"	

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

Добавление дочернего класса	
Атрибут класса	<input type="text"/>
Класс	<input type="text"/>
<input type="button" value="СОХРАНИТЬ"/> <input type="button" value="ОТМЕНА"/>	

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».

The screenshot shows a table with two columns: 'Наименование атрибута ↑' (Attribute Name) and 'Описание класса' (Class Description). The rows show 'childwindow' with 'Grid' and 'Grid Tree' descriptions, and 'childwindow' with 'Грид дерева для отображения данных' (Grid tree for displaying data) description.

Наименование атрибута ↑	Описание класса
childwindow	Grid
childwindow	Grid Tree Грид дерева для отображения данных

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»;

The screenshot shows the 'Objects' page with a search bar and a 'Create' button highlighted with a red box. Below is a table with columns: Найти (Search), Создать (Create), Описание (Description), Отображаемое имя (Visible name), Имя сервиса (Service name), Класс (Class), Используется на страницах (Used on pages), Метод модификации (Modification method), and Порядок сортировки (Sorting order). The table lists various objects like 'Wysiwyg panel', 'USPO Test_USPO_Report', 'Wysiwyg Module', etc.

Найти	Создать	Описание	Отображаемое имя	Имя сервиса	Класс	Используется на страницах	Метод модификации	Порядок сортировки
	+создать							
Корневой каталог								
Wysiwyg panel		Wysiwyg panel		Panel	WYSIWYG			
USPO Test_USPO_Report		Тестирование отчета УСПО	Тестирование отчета УСПО	USPOShowReportPage	Grid	Test USPO Report	pkg	
Wysiwyg		Страница с демонстрацией текст...			Wysiwyg Module	CORE-702		
Module Grid		Список модулей	Список модулей	GetModuleList	Grid	Модули	pkg_json_meta_f_modify_module	
Account Grid		Account Grid			Grid	Пользователи		

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»;

Наименование объекта

Используется на страницах
Объекты

+ создать

Наименование	Описание	Отображаемое имя	Имя сервиса	Класс	Используется на страницах
Корневой каталог					
SYS Grid Object << DO NOT CHANGE	Объекты метамодели	Grid Tree	MTObject	Grid Tree	Объекты
Редактировать значение	Атрибуты объекта	Атрибуты объекта	Attrib...	MTObjectAttribute	Grid

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

Редактирование Объекта

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

СОХРАНИТЬ **ОТМЕНА**

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»;

Наименование объекта		Используется на страницах	
Наименование		Описание	Отображаемое имя
<input checked="" type="checkbox"/> SYS Settings		L_sys_setting	Системные настройки экземпляра...
<input checked="" type="checkbox"/> USPO Authorization <<...		Список систем авторизации	MTGetSysSettings
<input checked="" type="checkbox"/> USPO Resource << DO ...		Список ресурсов	Grid
<input checked="" type="checkbox"/> USPO Source << DO ...		Источники данных	USPOShowAuthorization
<input checked="" type="checkbox"/> USPO Reports << DO ...		Отчеты	Grid
<input checked="" type="checkbox"/> TEST Print Report		TEST Print Report	Panel
<input checked="" type="checkbox"/> Test Grid		Грид "Test"	ShowReports
		Test	Grid
			test print
			pkg_json_report.f_modify_report
			test page

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»;

Клонирование объекта

Родительский объект	<input type="button" value="🔍"/>
Родительский объект	<input type="button" value="🔍"/>
<input checked="" type="checkbox"/> <input type="radio"/> <input type="radio"/>	
Наименование ↑	Описание объекта
Account Grid	Account Grid
Account Tabs	Account Tabs
Action Grid	Справочник действий
Action Tabs	Action Tabs
Menu Redirect Column Icon << Reusable	Menu Redirect Column Icon << Reusable
Module Grid	Список модулей
Roles Grid	Справочник ролей

4. нажать кнопку «Клонировать».

Клонирование объекта

Родительский объект	<input type="button" value="🔍"/>
	<input type="button" value="Клонировать"/>
Zначение по умолчанию	Zначение (переопределено)
true	

Важно!

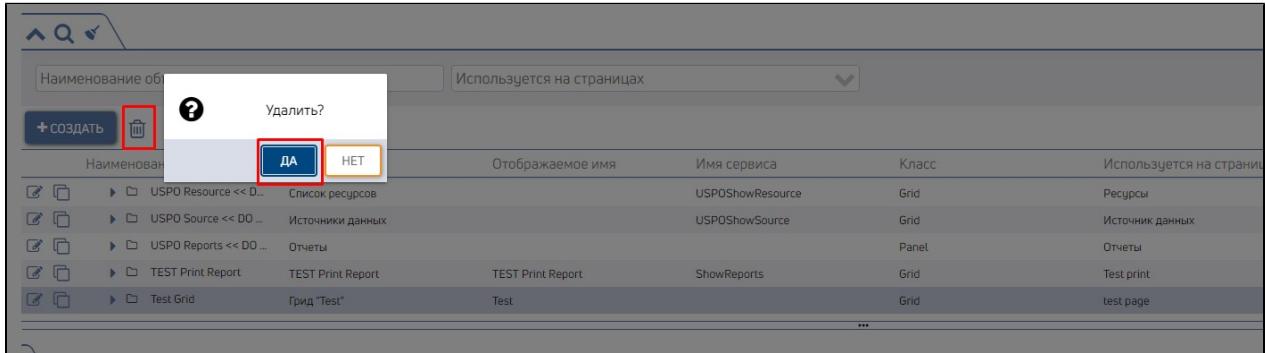
При клонировании объекта выполняется проверка на соответствие Иерархии классов.

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"/>	SYS Tab Page << DO ...	Таб-панель Страницы		Tab Panel	Страницы	1000 100
<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	pkg_json_scenario_f_modify_scen...
<input type="checkbox"/>	<input type="checkbox"/>	Scenario Step Grid	Шаги сценария	Шаги сценария	ATShowStep	Сценарии	pkg_json_scenario_f_modify_step...
<input type="checkbox"/>	<input type="checkbox"/>	Scenario Action Grid	Действия шага	Действия шага	ATShowAction	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	MTObject	Grid Tree	Объекты	pkg_json_meta_f_modify_object
<input type="checkbox"/>	<input type="checkbox"/>	SYS Grid Object Attrib...	Грид "Атрибуты объекта"	MTObjectAttribute	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	1000 900

...

Атрибуты объекта

Атрибуты объекта			
		Найменование	Значение по умолчанию
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	autoload	true
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Редактировать значение	ok_id
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	bnaudit	true
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	bndelete	true
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	btxcel	false
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	btrrefresh	true
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	checkboxidmore	true

3. ввести переопределенное значение атрибута объекта;

4. нажать кнопку «Сохранить».

<input type="checkbox"/>	<input type="checkbox"/>	Scenario Step Grid	Сценарии шагов	Шаги сценария	ATShowStep	Grid	Сценарии	pkg_json_scenario_f_modify_step
<input type="checkbox"/>	<input type="checkbox"/>	Scenario Action Grid	Действия шага	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	1000 100
<input type="checkbox"/>	<input type="checkbox"/>	SYS Grid Page Object ...	Объекты на странице	MTPageObject	Grid Tree	Страницы	pkg_json_meta_f_modify_page_obj...	1000 300
<input type="checkbox"/>	<input type="checkbox"/>	SYS Grid Object << DO...	Объекты метамодели	MTObject	Grid Tree	Объекты	pkg_json_meta_f_modify_object	1000 400
<input type="checkbox"/>	<input type="checkbox"/>	SYS Grid Object Attrib...	Грид "Атрибуты объекта"	MTObjectAttribute	Grid	Объекты	pkg_json_meta_f_modify_object_a...	1000 401
<input type="checkbox"/>	<input type="checkbox"/>	SYS Grid Class << DO...	Таблица с классами	MTClass	Grid	ввввв. Классы	pkg_json_meta_f_modify_class	1000 800
<input type="checkbox"/>	<input type="checkbox"/>	SYS Tab Panel Class ...	Панель со вкладками		Tab Panel	Классы	1000 900	

...

Атрибуты объекта

Атрибуты объекта			
		Найменование	Значение по умолчанию
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	autoload	true
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	autoselectidentity	ok_id
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	bnaudit	true
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	bndelete	true

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	1000 100
<input type="checkbox"/>	<input type="checkbox"/>	SYS Grid Page Object ...	Объекты на странице	MTPageObject	Grid Tree	Страницы	pkg_json_meta_f_modify_page_obj...	1000 300
<input type="checkbox"/>	<input type="checkbox"/>	SYS Grid Object << DO...	Объекты метамодели	MTObject	Grid Tree	Объекты	pkg_json_meta_f_modify_object	1000 400
<input type="checkbox"/>	<input type="checkbox"/>	SYS Grid Object Attrib...	Грид "Атрибуты объекта"	MTObjectAttribute	Grid	Объекты	pkg_json_meta_f_modify_object_a...	1000 401
<input type="checkbox"/>	<input type="checkbox"/>	SYS Grid Class << DO...	Таблица с классами	MTClass	Grid	ввввв. Классы	pkg_json_meta_f_modify_class	1000 800
<input type="checkbox"/>	<input type="checkbox"/>	SYS Tab Panel Class ...	Панель со вкладками		Tab Panel	Классы	1000 900	

...

Атрибуты объекта

Атрибуты объекта			
		Найменование	Значение по умолчанию
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	autoload	true
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	autoselectidentity	ok_id
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	bnaudit	true
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	bndelete	true

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;

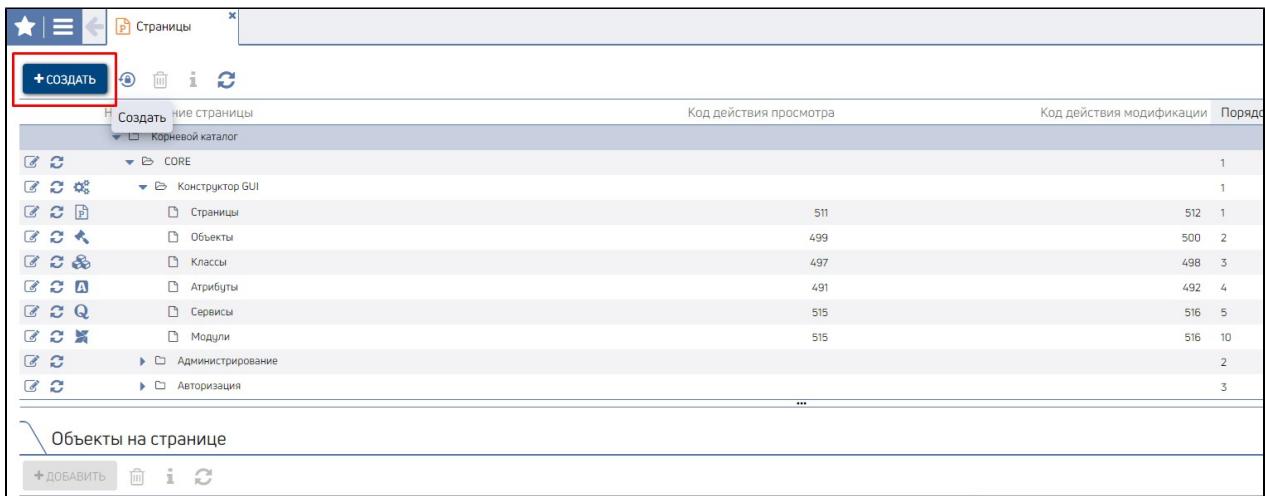
- Создание страницы
- Редактирование страниц
- Удаление страниц
- Сброс кэша
- Добавление объектов на страницу
- Редактирование объектов на странице

- 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»;



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:

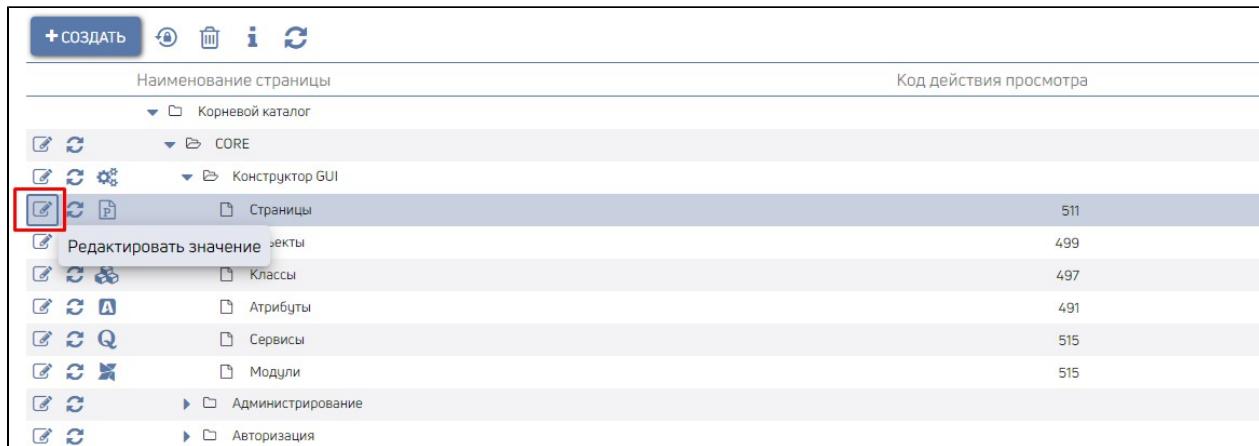
Наименование поля	Тип данных	Комментарии
Наименование	Текст	Обязательно для заполнения.
Тип	Выбор из списка	Обязательно для заполнения.
Порядок	Число	Обязательно для заполнения.
Иконка	Выбор из списка	
Код действия просмотра	Число	Заполняется только для типа «Страница».
Код действия модификации	Число	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	<p>Адрес статической страницы.</p> <p>Заполняется только при значении поля «Статическая страница» = «Да».</p> <p>Доступные символы: [0-9],[a-z],[A-Z],[_],[-].</p> <p>Например, при указании значения "test" ссылка на страницу будет: http://адрес/page/test</p>

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 management interface. The 'Edit value' button (highlighted with a red box) is located in the header of the table. The table has columns: 'Наименование страницы' (Page Name), 'Код действия просмотра' (View Action Code), and a list of pages. One page is selected: 'Страницы' (Pages) with code 511.

Наименование страницы	Код действия просмотра
Страницы	511
Редактировать значение зекты	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 management interface. The 'Delete' button (highlighted with a red box) is located in the header of the table. The table has columns: 'Наименование страницы' (Page Name), 'Код действия просмотра' (View Action Code), and a list of pages. One page is selected: 'Страницы' (Pages) with code 511.

Наименование страницы	Код действия просмотра
Страницы	511
Редактировать значение зекты	499
Классы	497
Атрибуты	491
Сервисы	515
Модули	515
Администрирование	
Авторизация	

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.

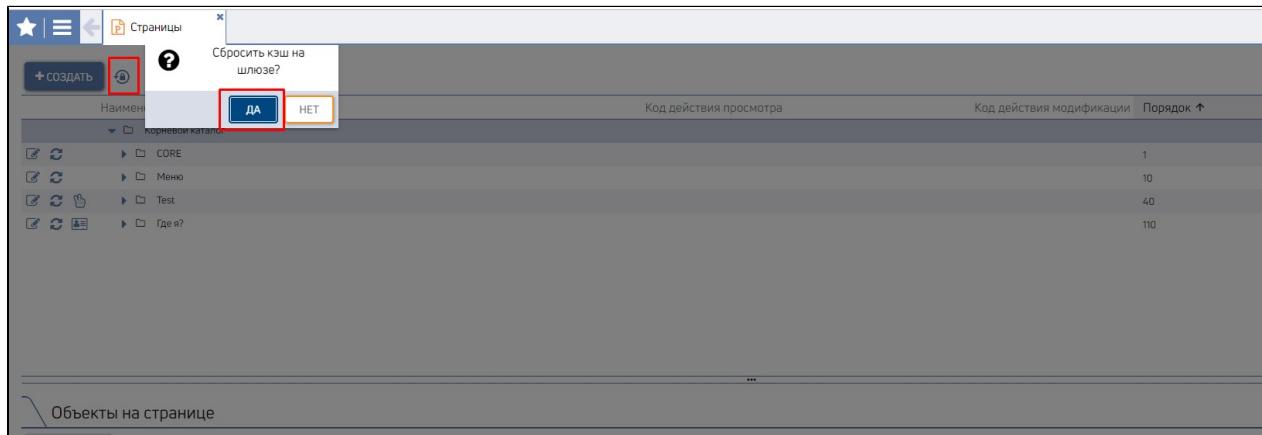
Сброс кэша

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»;

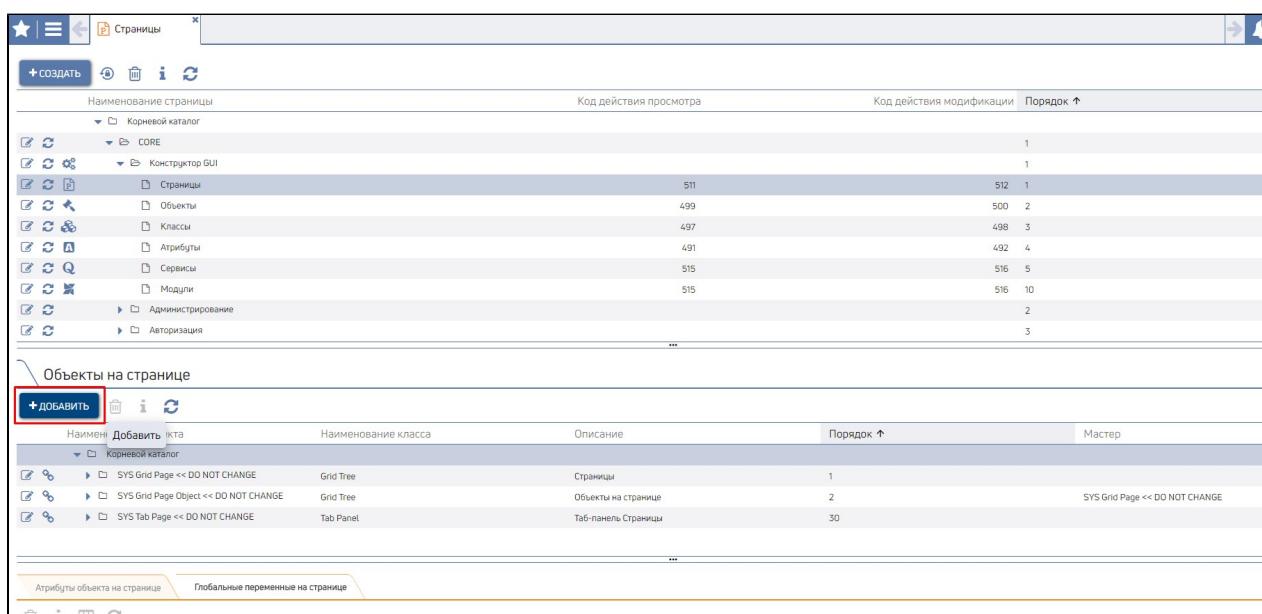


4. дождаться завершения технологических работ, связанных со сбросом.

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 зависимости 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:

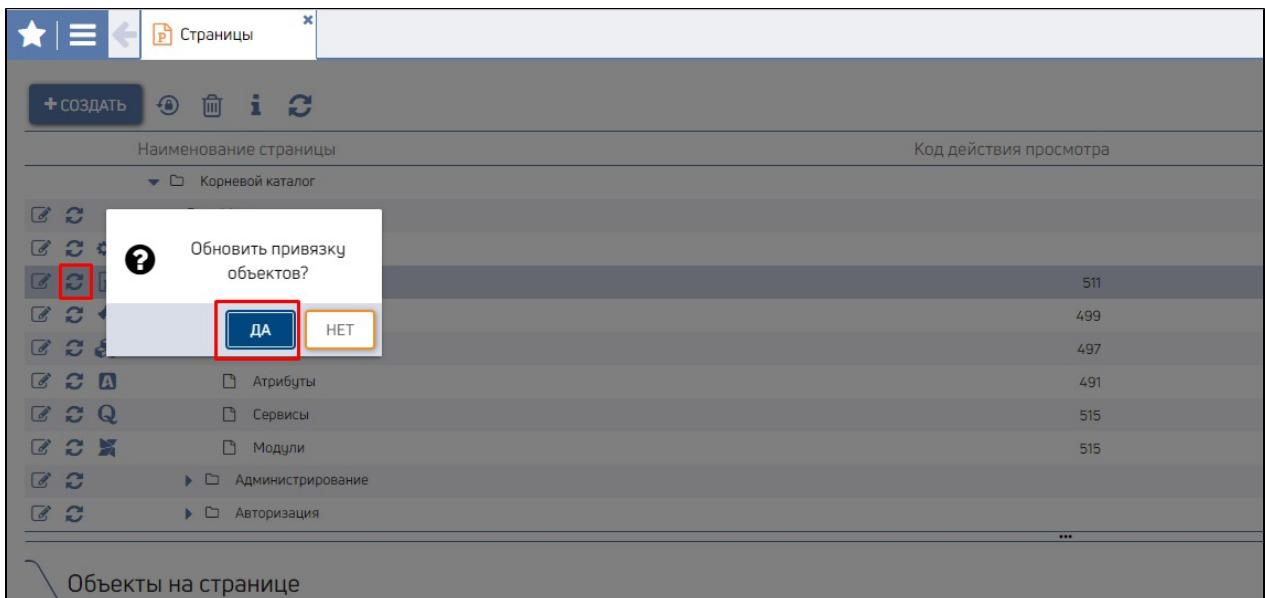
Наименование	Описание
disabledemptyaster	Признак блокировки, если мастер вернул пустое значение
getmastervalue	Наименование параметра из мастера, который будет передан в json в виде "master": {"наименование_параметра":"значение"}
reloadmaster	Признак обновления родителя по мастеру
reloadservice	Признак повторного вызова сервиса комбобокса во время раскрытия в случае, если изменился мастер
reqsel	Признак зависимости от мастера (true/false)

Обновление привязки объектов на странице

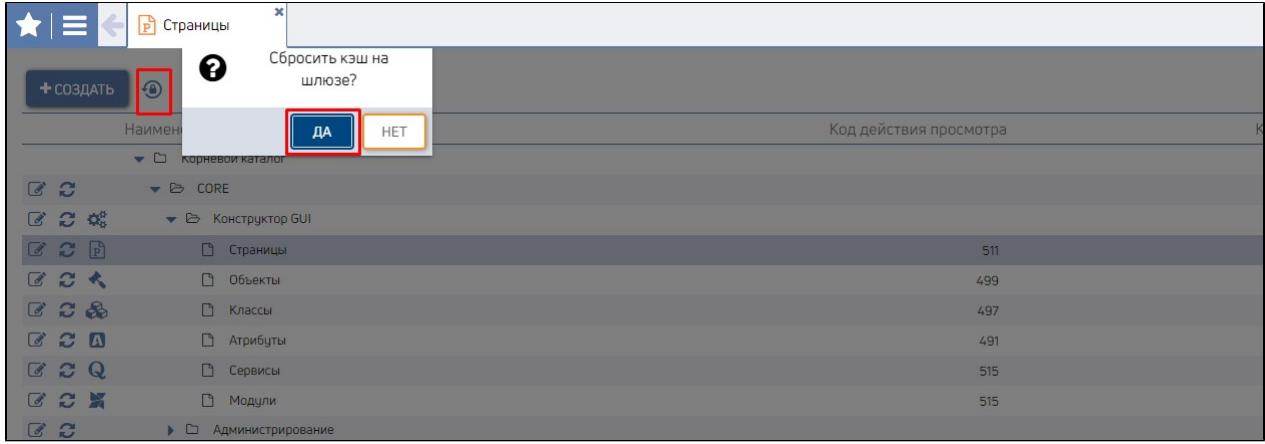
В случае, если в объект, который уже используется на странице, были внесены изменения (например, добавлены новые дочерние объекты) необходимо обновить привязку объектов на странице. В этом случае все внесенные изменения отобразятся на странице.

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»;

Наименование объекта	Наименование класса	Отображаемое имя	Описание	Порядок ↑	Мастер
Корневой каталог				1	
SYS Grid Class << DO NOT CHANGE	Grid	Таблица с классами		1	
SYS Tab Panel Class << DO NOT C...	Tab Panel	Панель со вкладками		2	SYS Grid Class << DO NOT CHANGE

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

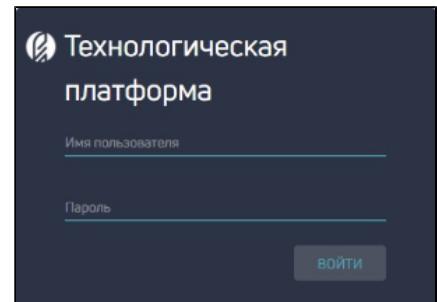
Пользовательский интерфейс (графический интерфейс) Конструктора состоит из стандартных элементов, таких как: кнопки, меню, вкладки, фильтры, панели навигации и управления и т.п.

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



Символы введённого пароля будут скрыты.

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».

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.

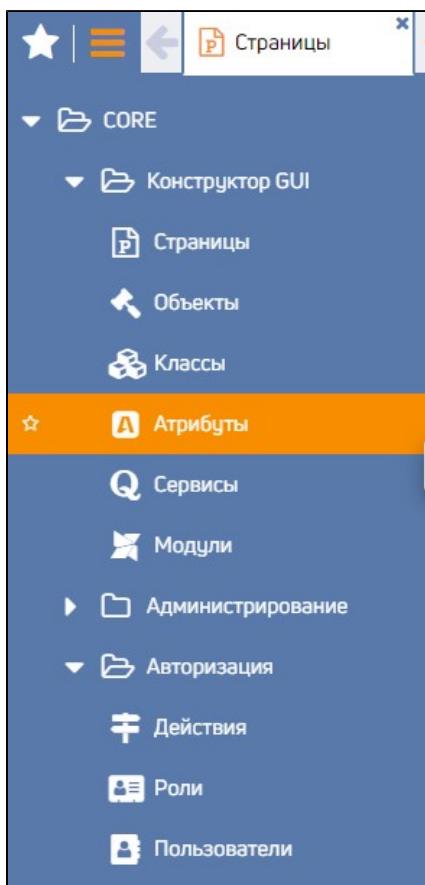
6. Кнопка вызова профиля пользователя

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

7. Кнопка вызова информации о программе

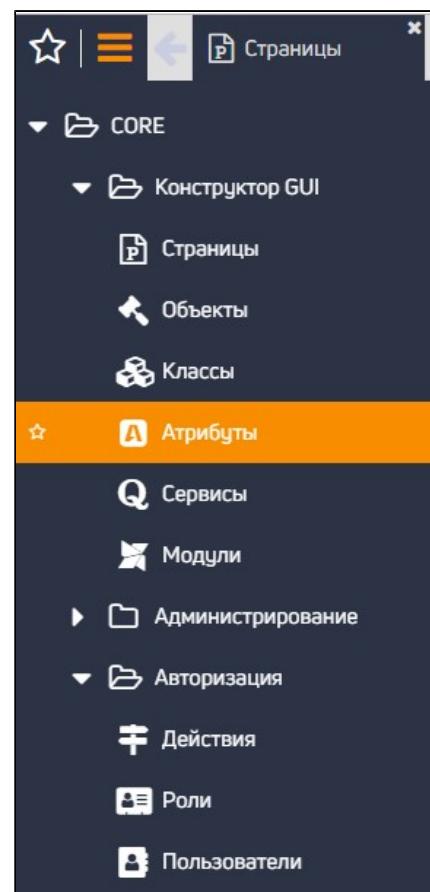
Кнопка вызывает информацию о версии программы и разработчике.

Основное меню, панель закладок (быстрого доступа)

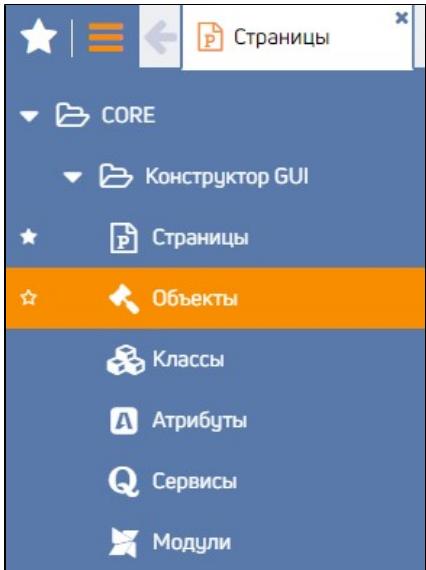


Основное меню визуально представляет собой дерево каталогов.

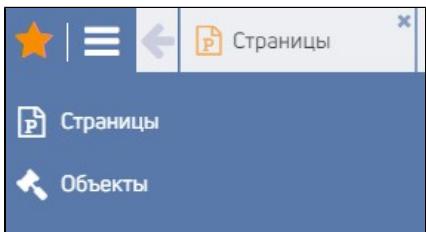
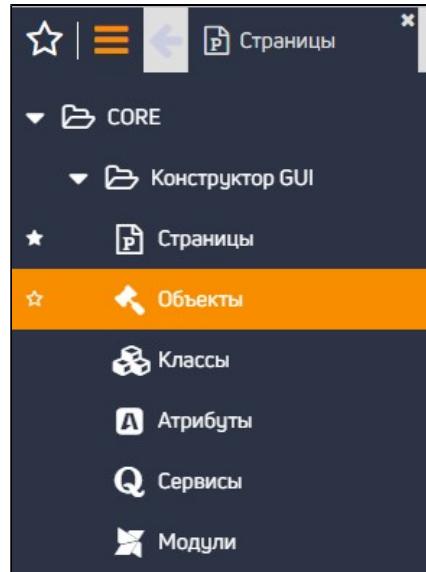
Каталог раскрывается при нажатии на название. Если в каталоге вложены ещё каталоги, то раскрытие будет происходить до достижения конечной страницы. Выбор страницы также происходит нажатием на название. Каталог имеет значок «Папка», при раскрытии каталога значок меняет свой вид на «раскрытый», а стрелка,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



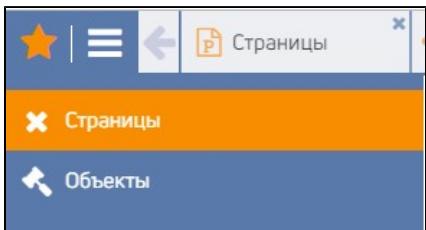
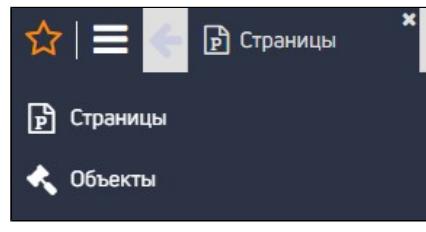
Для добавления страниц в панели быстрого доступа «Избранное» необходимо навести курсор на страницу, слева появится значок «звёздочка». Нажать на этот значок. «Звёздочка» выделится и будет постоянно отображаться напротив названия страницы в основном меню. Для снятия значка необходимо нажать



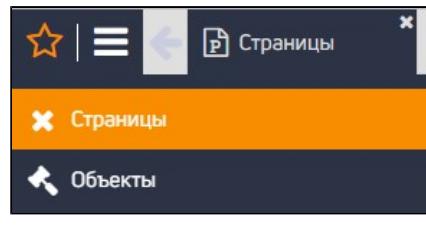
на него повторно. При снятии значка, страница будет удалена из панели «Избранное»



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».

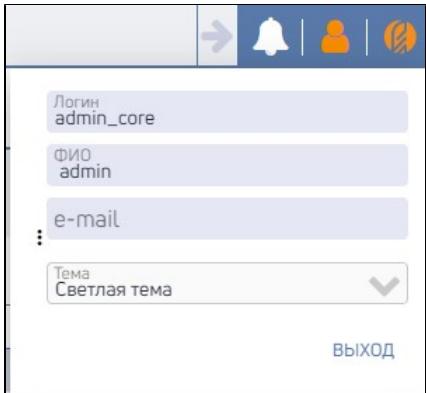


Здесь содержится информация для разработчиков, системного администратора и технической поддержки. Здесь можно переключаться по вкладкам и типам оповещений.

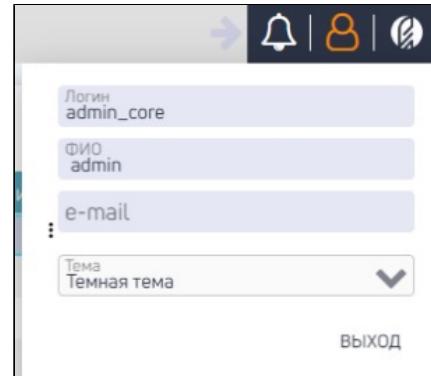
Распределение информации по блокам:

- Все - все полученные сообщения независимо от типа и оповещения из БД;
- Информация - информационные сообщения (тип info);
- Ошибки - сообщения об ошибках (тип error);
- Предупреждения - предупреждения, требующие подтверждения пользователя (тип warning);
- Оповещения - сообщения, полученные от службы оповещения;
- Разработка - сообщения отладки и список глобальных переменных для каждой открытой страницы.

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.

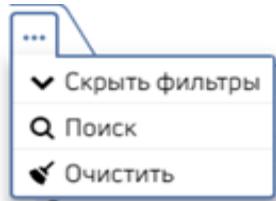
Элементы управления и их обозначение

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



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



Поиск. Ключевой

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

		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.	Информация Печать в excel	
		Собиратель кнопок для периодов. Элемент предназначен для скрытия обобщения процедур закрытия/открытия расчетных периодов.	Закрыть Расчетный период Закрыть Финансовый период	
		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 исторических 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 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.

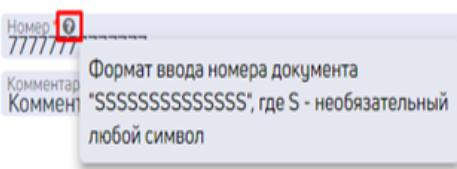
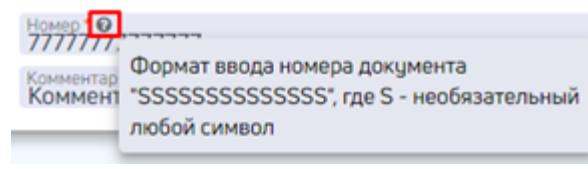
Необязательные поля могут быть заполнены при наличии информации или по мере её поступления.

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	

<div style="border: 1px solid #ccc; padding: 5px; display: flex; align-items: center;"> Организация </div> <p>Поле фильтрации. Для внесения данных в такие поля, необходимо нажать на значок лупы и выбрать запись из выпадающего фильтра</p>	
<div style="border: 1px solid #ccc; padding: 5px; display: flex; align-items: center;"> Наименование </div> <p>Простое поле. В поле осуществляется ввод текстовой или числовой информации.</p>	<div style="border: 1px solid #ccc; padding: 5px; display: flex; align-items: center;"> Наименование </div>
<div style="border: 1px solid #ccc; padding: 5px; display: flex; align-items: center;"> Период * </div> <p>Поле даты. В поле осуществляется ввод даты в заданном формате (например, ДД.ММ.ГГГГ) или из выпадающего календаря при нажатии на значок.</p>	<div style="border: 1px solid #ccc; padding: 5px; display: flex; align-items: center;"> Период * </div>
<div style="border: 1px solid #ccc; padding: 5px; display: flex; align-items: center;"> Тип </div> <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>	<div style="border: 1px solid #ccc; padding: 5px; display: flex; align-items: center;"> Тип </div>
<div style="border: 1px solid #ccc; padding: 5px; display: flex; align-items: center;"> УO не указана </div> <p>Чекбокс. Атрибут, выполняет логическую функцию «да /нет», в соответствии с подписью</p>	<div style="border: 1px solid #ccc; padding: 5px; display: flex; align-items: center;"> УO не указана </div>
<div style="border: 2px solid #0070C0; padding: 5px; display: flex; align-items: center;"> Адрес </div> <p>Поле быстрого перехода. Такие поля позволяют осуществить быстрый переход к</p>	<div style="border: 2px solid #0070C0; padding: 5px; display: flex; align-items: center;"> Адрес </div>

	другой странице с автоматическим внесением имеющихся в поле данных	
	<p>Подсказка. Располагается непосредственно в поле для заполнения. При наведении курсора на значок подсказки появляется информация по правилам заполнения данного поля</p>	

Окна для ввода данных

Ввод и редактирование данных для таблиц проходит 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

Добавить ещё
СОХРАНИТЬ
ОТМЕНА

Добавление

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 догово...	Поставщик
01.06.2018		Электроснабжение т...	Электроснабжение т...	ИВМ Кратков1
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)
 - Панель с шагами (Panel Road Map)
- Поля
 - Встроенный фрейм (IFrame)
 - Групповые поля
 - Field Group
 - Field Set
 - Поле "Выпадающая таблица" (Field Grid)
 - Поле "Выпадающая таблица-дерево" (Field Grid Tree)
 - 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

Объекты классов Кнопка (Button) либо Собиратель кнопок (Button/Collector) со значением `cn_order < 1 000 000` будут отображены **до** блока с кнопками по умолчанию.

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. `btdaudit` – кнопка «Информация»;
3. `btdexcel` – кнопка «Экспорт в Excel»;
4. `btdrefresh` – кнопка «Обновить».

Для всех статических кнопок порядок сортировки (`cn_order`) = 1 000 000.

Объекты классов Кнопка (Button) либо Собиратель кнопок (Button/Collector) со значением `cn_order < 1 000 000` будут отображены **до** блока с кнопками по умолчанию.

Объекты классов Кнопка (Button) либо Собиратель кнопок (Button/Collector) со значением `cn_order > 1 000 000` будут отображены **после** блока с кнопками по умолчанию.

Сортировка данных в гриде настраивается с помощью атрибутов:

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

Настройка работы со значениями грида производится с помощью атрибутов:

Наименование	Описание
--------------	----------

атрибута	
collectionvalues	Тип сбора значений: object - виде объекта; array - массив строк.
selmode	Возможность выделения нескольких значений в GRID и TREEGRID SINGLE - только 1 значение SIMPLE - позволяет 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 сплитера.

Простой грид (Grid)

Представляет собой таблицу, отображающую данные в строках и столбцах.

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	Сценарии: Scenario Grid...
ATShowScenario	meta	select	po_session	Сценарии: Scenario Step Grid
ATShowStep	meta	select	po_session	Сценарии: Scenario Step Grid
GetCurrentDate	meta	select	po_session	
GetMetamodePage	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;

СОХРАНИТЬ		ОТМЕНА	Редактирование
Наименование	Описание	Признак выводимого набора данных	Признак самостоятельного заполнения
<input checked="" type="checkbox"/> Box	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 *	<input type="text"/>
Сервис	<input type="text"/>
Имя объекта *	<input type="text"/>
Описание объекта *	<input type="text"/>
Порядок сортировки *	<input type="text"/>
Отображаемое имя	<input type="text"/>
Метод модификации	<input type="text"/>
Провайдер данных при модификации	<input type="text"/>

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 режим модификации 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, idproperty, valuefield, сервис)
4. указать необходимый режим добавления/редактирования в атрибуте editmode поля Combobox.

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 на колонке.

5. Наименование поля в автособираемом окне будет браться из наименования колонки.

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.

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

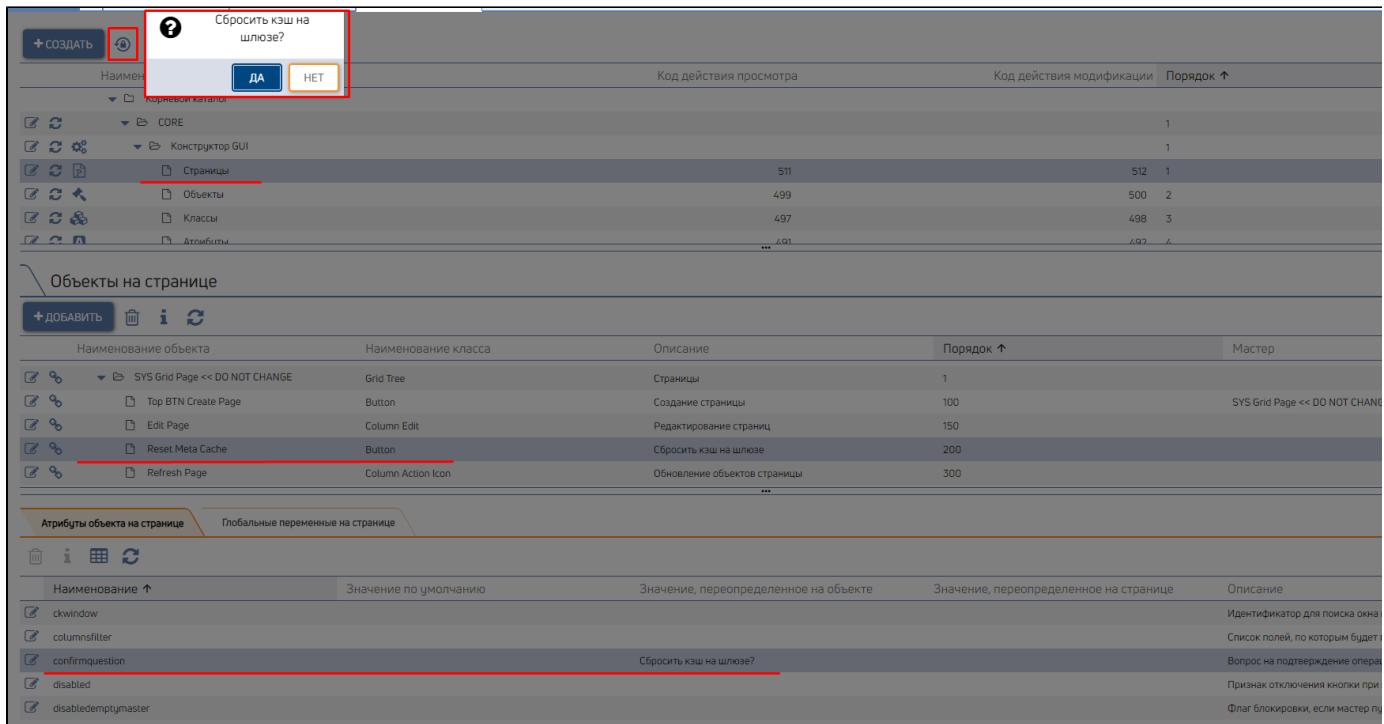
Для корректного отображение иерархической структуры грида типа "Дерево" в результирующем наборе сервиса должны быть следующие ключ-значения:

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 ckwindow. Для настройки связи необходимо указать одинаковое значение атрибута ckwindow 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 выгрузки файла
8	Режим загрузки файла

- modeaction - Значение action, передаваемое 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 - Отображается только иконка кнопки (true/false)

tipmsg - Подсказка у кнопки при наведении

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

Светлая тема	uitype	Темная тема
	1	
	2	

Отображение кнопок при инициализации настраивается с помощью атрибутов:

- disabled - признак неактивности кнопки

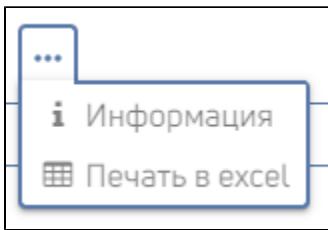
- disabledemptymaster - признак неактивности кнопки при пустом мастере
 - hidden - признак скрытия кнопки
-

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 Информация и Экспорт 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', 'Страницы', and 'Объекты'. Below the navigation bar is a tree view of objects under 'Конструктор GUI': 'Страницы', 'Объекты', 'Классы', and 'Атрибуты'. A table lists object details like ID, Name, Description, Order, and Master. The 'Объекты на странице' tab is selected, showing a list of objects with their names, descriptions, and masters. The 'Атрибуты объекта на странице' tab is also visible at the bottom. The 'Visible' attribute is set to 'true'.

Наименование объекта	Наименование класса	Описание	Порядок ↑	Мастер
вокруг	Window	Кнопка "Редактировать" в гриде	150	SYS Grid Object << DO NOT CHANGE
Add/Edit Object Window	Window	Объект	180	
Clone Object Window	Window	Окно клонирования объекта	250	SYS Grid Object << DO NOT CHANGE
Field Parent Object	Field Tree	Родительский объект	100	
Column Name	Column Tree	Колонка Имя	100	
Column Description	Column String	Описание объекта	200	
Button Clone Object	Button	Клонировать объект	200	SYS Grid Object << DO NOT CHANGE

Наименование ↑	Значение по умолчанию	Значение, переопределенное на объекте	Значение, переопределенное на странице	Описание
setglobal				Имя глобальной переменной, должна начинаться с g
sortcolumn		cv_name_lowered		Колонка по которой будет сортировать грид вместо column
stylerules				Собирает отображения styles для колонок...
visibleinwindow	false			Признак отображения в автосохраняющем окне неактивны...
visible	true			Признак отображения колонки

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.

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		Удалить	Информация	Список	Обновить
	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 ↑

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 его необходимо скрыть для итоговой строки.

		Наименование услуги по договору ↑
	▶	AP Вода для домашних животных
	▶	Водоотведение зебра
	▶	Накопительный фонд (по проживающим)
	▶	Отопление (по отапливаемой площади)
	▶	Тест
	▶	XBC 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:

+Создать

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	SYS Grid Page <<
Top BTN Create Page	Button	Создание страницы	100	
Edit Page	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 currency sign 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.

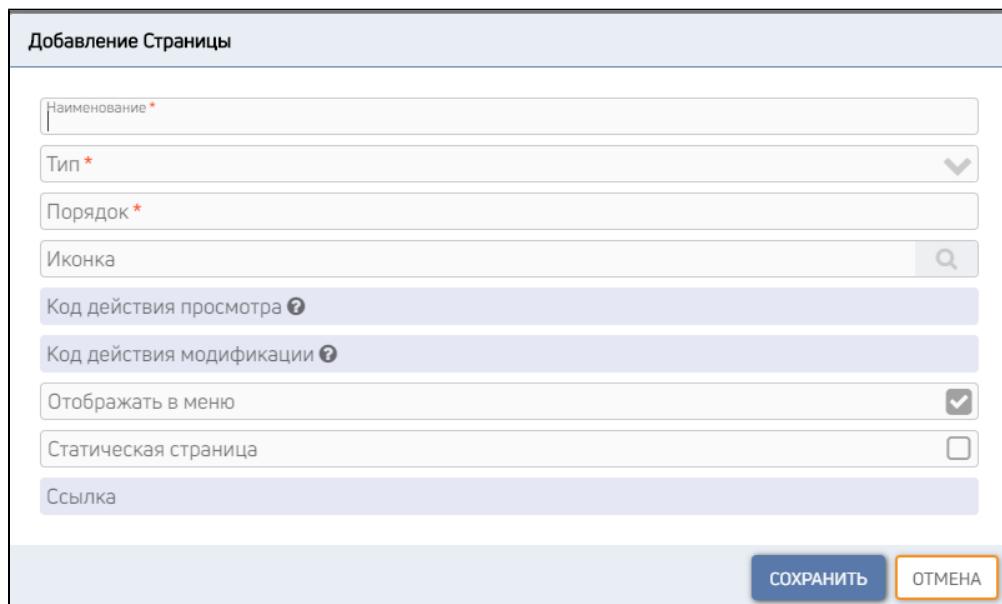
В атрибуте thousandseparator - разделитель тысяч.

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 Окно 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 при нажатии на кнопку в гриде с handler = 'onCreateChildWindowMaster' либо handler = 'onRowCreateChildWindowMaster'. Включает в себя поля, сформированные 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 ckwwindow у окна и кнопки (Button) в гриде не совпадает.

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

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, idproperty, valuefield, сервис)
4. указать необходимый режим добавления/редактирования в атрибуте editmode поля Combobox.

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 на колонке.

5. Наименование поля в автособираемом окне будет браться из наименования колонки.

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 историчными данными (Panel History)
- Панель со вкладками (Tab Panel)
- Панель с фильтрами (Filter Panel)
- Панель с шагами (Panel Road Map)

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.

Динамическая панель будет иметь стандартный заголовок вида:

```
{  
    "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": "_",  
    "childs": [ '  
        { }  
    ] }';  
--
```

Лайфхак

Для формирования корректного JSON можно:

1. на тестовой странице нарисовать объект, который хотим возвращать с бэка.
2. открыть страницу с нарисованным объектом.
3. скопировать структуру JSON нужного объекта из сервиса GetMetamodelPage

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": "_",  
    "childs": [ '  
        { }  
    ] }';  
--
```

```

"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) || ''",
"child": [{"
"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": " ",
        "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",
"childs": [
{
"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,
"IFIELD",
"visibleinwindow": "false"
},
{
"cl_dataset": "0",
"cv_displayed": " \\"\\",
"cv_description": " \\\"\\",
"ck_page": "301",
"column": "cl_overstep",
"fieldlabel": "&nbsp;",
"width": "
```

```

        "40%",
        "datatype": "checkbox",
        "checked": true,
        "type": "IFIELD",
        "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.

The screenshot shows a user interface with two main sections highlighted by red boxes:

- Left Section (Representative Details):**
 - Документ для действий реквизитов * (Document for document actions) - dropdown menu showing "01.02.2019".
 - Фамилия руководителя * (Last name of the head) - input field with placeholder "Иванов Иван Иванович".
 - Сертификат руководителя * (Head certificate) - input field with placeholder "Иванов Иван Иванович".
 - Имя руководителя * (First name of the head) - input field with placeholder "Иван Иванович".
 - Отчество руководителя (Middle name of the head) - input field.
 - Документ удостоверяющей личность (Document certifying identity) - dropdown menu showing "Свидетельство о предоставлении временного убежища на территории Российской Федерации".
 - Серия * (Series) - dropdown menu showing "KK".
 - Номер * (Number) - input field with placeholder "1234567890".
 - Комментарий (Comment) - text area.
- Right Section (Organizational Contact Information):**
 - ФИО контактного лица (Name of the contact person) - input field.
 - Город организаций (City of the organization) - dropdown menu showing "111 111-1111".
 - Город телефона (City of the phone) - dropdown menu showing "Городской".
 - Электронная почта организации (Organization's email) - input field.
 - Факс организации (Fax of the organization) - input field.
 - Сайт организации (Organization website) - input field.
 - Телефон документической службы (Phone of the documentation service) - input field.
 - (333) 333-33-33
 - Режим работы (Working hours) - input field.

Контейнер типа «Панель» (Panel)

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

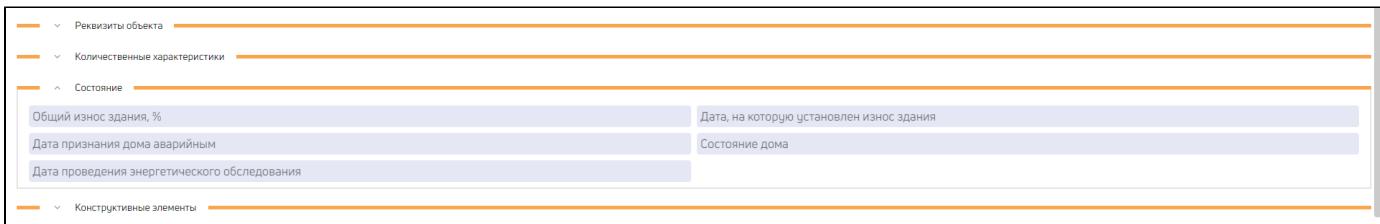
- Кнопки и Собиратели кнопок;
- Фильтры;
- Гриды (простые и гриды-дерево);
- Поля (текстовые, числовые и т.д.);
- Панели (Панели со вкладками, ПИД, контейнеры типа «Панель» и т.д.).

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:



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).

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).

Атрибут align определяет расположение наполнения Панели.

Доступные значения:

- left - Объекты выровнены по левой стороне Панели;
- right - Объекты выровнены по правой стороне;
- center - Объекты выровнены по центру.

Если для панели не задано значение align, то поля растягиваются по всей ширине контейнера.

Если align задан, то поля принимают определенную ширину для того, чтобы сгруппироваться слева, справа или по центру.

Панель детализации в гриде (Panel Detail)

Позволяет выводить поля детализации к записи в гриде.

Наименование услуги по договору ↑	Признак ОДН	Сальдо на начало меся...	Начислено	Ручная корректировка	Оплачено
AP Вода для домашних животных	Договор 1-1		Подразделение КО "Соколиная гора"		
Поставщик ООО "АРИАЛ"					
▶ 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.

Кнопки "Добавить", "Отмена", "Удалить", "Скачать" и "Добавить файл" - захардкожены в классе Panel File.

Для корректной пользовательской настройки указанных кнопок необходимо создать в Panel File дочерние объекты класса Button с соответствующим наименованием объекта:

Кнопка "Отмена" при добавлении файла	Override Cancel Button
Кнопка "Добавить файл" при добавлении файла	Override Save Button
Кнопка "Удалить"	Override Delete Button
Кнопка "Скачать"	Override Download Button
Кнопка "Добавить" на панели просмотра файлов	Override Add Button

Последующая настройка проводится уже для Override-кнопок.

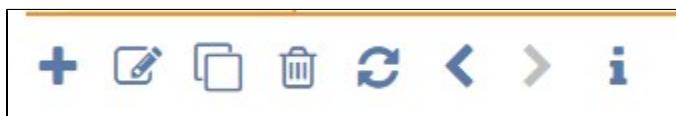
Панель с историчными данными (Panel History)

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

The screenshot shows a complex administrative form titled 'Паспорт дома' (House Passport). The top navigation bar includes tabs for 'Паспорт дома', 'Способ формирования ФКР', 'Договоры по объекту обслуживания', and 'Документы'. The main area is divided into several sections:

- Rekvizity obekta (Object Requisites):** Contains fields for 'Дата начала действия паспорта' (Passport validity start date) set to '11.05.2016', 'Класс энергетической эффективности' (Energy efficiency class), 'Код ОКТМО' (OKTMO code), 'Номер дома в ГИС ЖКХ' (House number in GIS), 'Серия дома' (House series), 'Код УСЭН' (USEN code), 'Часовой пояс' (Time zone) checked, and 'Является домом блокированной застройки' (Is a restricted construction house) unchecked.
- Kolichestvennye charkakteristiki (Quantitative Characteristics):** Contains fields for 'Общий износ здания, %' (General building wear, %), 'Дата признания дома аварийным' (Date of recognizing the building as dangerous), and 'Дата проведения энергетического обследования' (Date of energy audit).
- Konstruktivnye elementy (Constructive Elements):** A section at the bottom with a single field.

Управляющим элементом содержит 8 кнопок:



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

Для всех статических кнопок порядок сортировки (сп_order) = 1 000 000.

Объекты классов Кнопка (Button) либо Собиратель кнопок (Button/Collector) со значением сп_order < 1 000 000 будут отображены **до** блока с кнопками по умолчанию.

Объекты классов Кнопка (Button) либо Собиратель кнопок (Button/Collector) со значением сп_order > 1 000 000 будут отображены **после** блока с кнопками по умолчанию.

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 историчных 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 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.



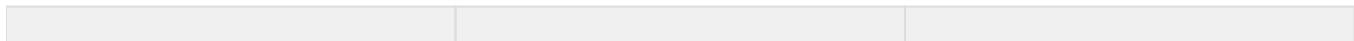
Панель со вкладками может включать в себя следующие типы объектов:

- Гриды (Простой грид, Грид-дерево);
- Контейнер типа «Панель»;
- Панель с историчными 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.

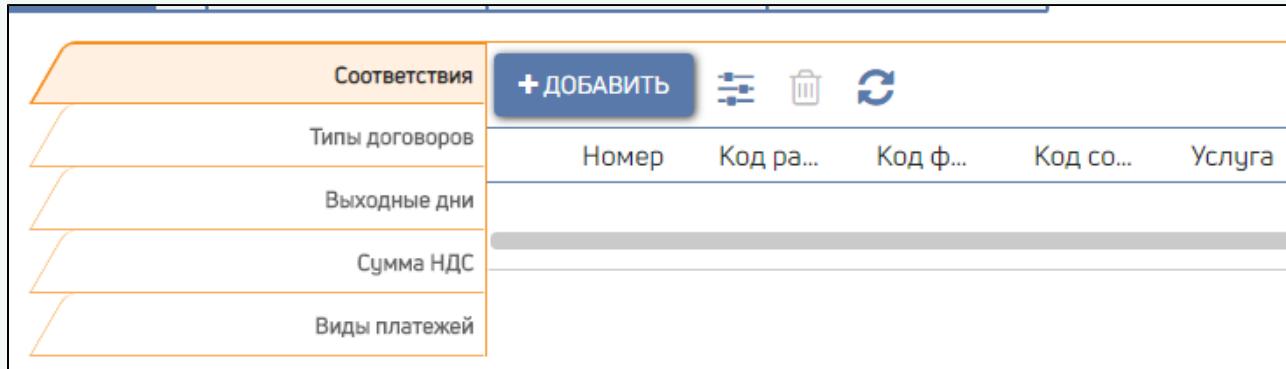
Отображение вкладок Tab Panel настраивается с помощью атрибутов:



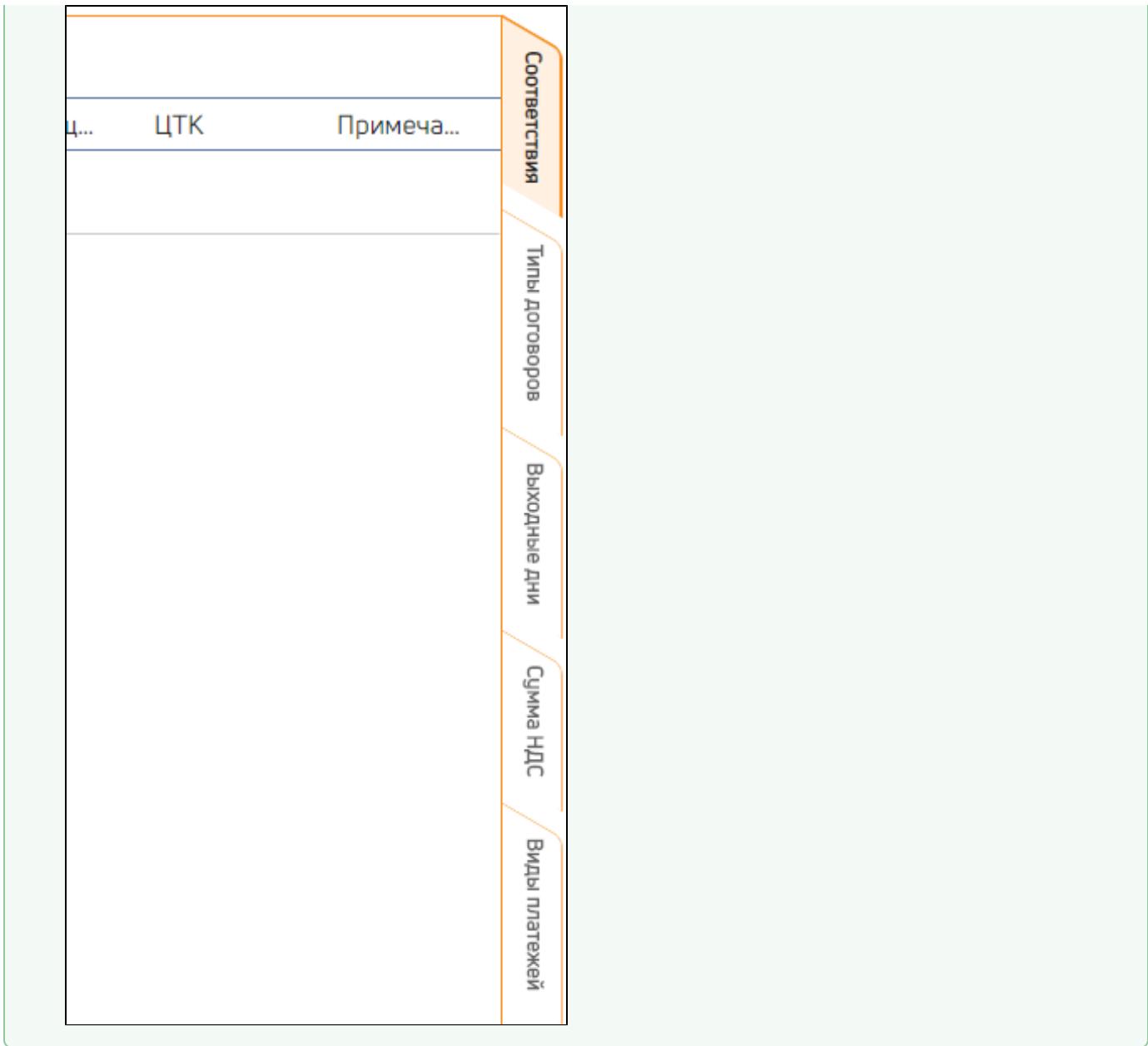
Атрибут	Доступные значения	Комментарий
align	left - слева; center - сверху по центру; right - справа	Расположение вкладок
contentview	hbox - горизонтально; vbox - вертикально	Направление текста в наименовании вкладок

Пример

align = left, contentview = hbox



align = right, contentview = vbox



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:

и Фильтр выглядит следующим образом:



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

Отключить сохранение фильтров можно с помощью атрибута `filtervaluesave = false`.

Панель с шагами (Panel Road Map)

Компонент для реализации форм с пошаговым заполнением.

Для создания шага в PRM необходимо создать дочерний объект класса Panel. В данный объект помещаем наполнение шага. Например, Грид, поля ввода/отображения, другую панель и тд.

Также PRM может содержать объекты класса Кнопка (Button) для переопределения статических кнопок:

- "Override Back Button" - для кнопки "Назад";
- "Override Cancel Button" - для кнопки "Отмена";
- "Override Next Button" - для кнопки "Далее";
- "Override Save Button" - для кнопки "Сохранить".

Переопределить кнопки можно как в самой PRM, так и в дочерних панелях.

Если объекты переопределены в дочерней панели, то правила переопределения действуют только на ней. Если переопределение выполняется на уровне PRM, то правила действуют для соответствующих объектов всех дочерних панелей PRM.

Поля в дочерних панелях PRM сразу будут отображаться в режиме редактирования.

Кнопки, размещенные в дочерних панелях с полями ввода не будут отображены.

Для каждого шага в PRM проставляются глобальные переменные:

- "gIsEnd" - признак что панель последняя из доступных;
- "gIsStart" - признак что панель начальная из доступных;
- "gPageNum" - отображаемый номер шага;
- "gPageIndex" - индекс шага (нумерация начинается с 0).

Для использования указанных глобальных переменных необходимо добавить их в список глобальных переменных на странице (CORE Конструктор GUI Страницы).

Обратите внимание!

Для корректной работы с указанными глобальными переменными необходимо верно указывать их наименование:

- "gIsEnd" - признак что панель последняя из доступных;
- "gIsStart" - признак что панель начальная из доступных;
- "gPageNum" - отображаемый номер шага;
- "gPageIndex" - индекс шага (нумерация начинается с 0).

Для изоляции к наименованию глобальной переменной добавляется `ck_id` объекта на странице (из `t_page_object`) класса Panel Road Map.

Пример

```
gIsEndF76E1A6689114274Aff0D956C4B90561: false  
gIsStartF76E1A6689114274Aff0D956C4B90561: true  
gPageNumF76E1A6689114274Aff0D956C4B90561: 1  
gPageIndexF76E1A6689114274Aff0D956C4B90561: 0
```

Поля

Дочерние элементы для контейнеров, окон и некоторых типов колонок.

- Встроенный фрейм (IFrame)
- Групповые поля
 - Field Group
 - Field Set
- Поле "Выпадающая таблица" (Field Grid)
- Поле "Выпадающая таблица-дерево" (Field Grid Tree)
- 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).

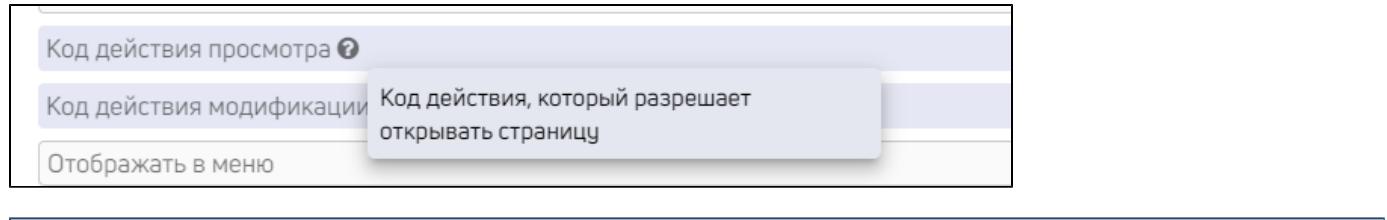
В атрибуте defaultvaluequery указывается наименование сервиса, возвращаемое значение которого и будет являться значением по умолчанию для поля.

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).

Встроенный фрейм (IFrame)

Поле для отображения сторонних ресурсов по URL либо HTML.

Размеры поля настраиваются с помощью атрибутов:

Наименование атрибута	Описание
height	Статическая высота в пикселях (px)
width	Ширина поля. Целое число от 1% до 100%. Обязательно добавлять %.

Формат возвращаемого сервисом значение указывается в атрибуте typeiframe. Возможные значения: "URL" и "HTML".

Для корректного отображения контента необходимо:

1. создать объект класса Iframe
2. привязать сервис, который будет возвращать либо URL, либо HTML контента
3. в атрибуте column для объекта указать наименование параметра сервиса, в котором возвращается URL/HTML
4. в атрибуте typeiframe указать тип возвращаемого сервисом значения
5. при необходимости указать высоту (height) и ширину (width).

Групповые поля

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

Field Group

Компонент для объединения нескольких полей в одну группу.

Позволяет настраивать поведение всей группы.



Визуализация компонента настраивается с помощью атрибутов:

Наименование атрибута	Описание
contentview	Вид наполнения: hbox: горизонтальное hbox-wrap: горизонтальное с переносом на следующую строку vbox: вертикальное
width	Ширина поля. Целое число от 1% до 100%. Обязательно добавлять %.

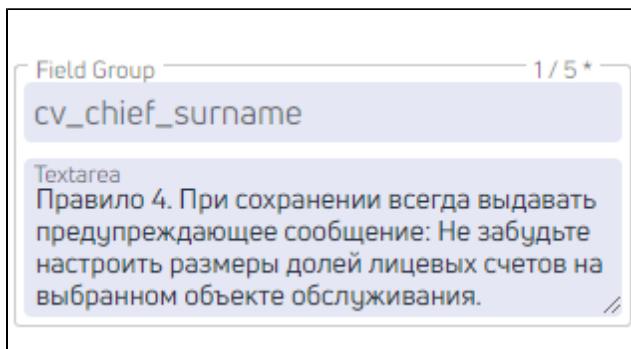
Блокировка/скрытие группы настраиваются с помощью стандартных атрибутов поля disabled/hidden соответственно.

Правила блокировки/скрытия указываются в атрибутах disabledrules/hiddenrules.

Атрибуты, позволяющие регулировать правила обязательности заполнения полей в группе:

Наименование атрибута	Описание
reqcount	Количество обязательно заполненных полей в группе, требуемое для корректной валидации
reqcountrules	Правила для изменения количества обязательно заполненных полей в группе. Имеет приоритет над reqcount. Используется синтаксис условного оператора: <i>условие ? выражение1 : выражение2</i>

При указании количества обязательно заполненных полей в группе в правом верхнем углу отображается счетчик вида: "сколько полей заполнено / сколько нужно заполнить".



Field Set

Данный компонент позволяет объединить несколько полей в один массив, также повторять данный массив n-раз.

Яркий пример использования [Динамическая панель](#)

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

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	Наименование параметра, который будет отображен	ключ-значение, которое будет отображено в интерфейсе
idproperty	Наименование параметра, отвечающего за уникальность записей	ключ-значение, которое содержит уникальный идентификатор записи
valuefield	Наименование параметра из внутреннего сервиса, значение которого будет передано для дальнейшей обработки	ключ-значение из сервиса Field Grid, из которого берется значение

Настройка работы со значениями грида производится с помощью атрибута:

Наименование атрибута	Описание
collectionvalues	Тип сбора значений: object - виде объекта; array - массив строк.

Настройка размера выпадающего грида настраивается с помощью атрибутов:

Наименование атрибута	Описание
pickerheight	Высота выпадающей таблицы/списка. По умолчанию 390
pickerwidth	Ширина выпадающей таблицы/списка

Настройка мультивыбора

1. Добавить в Field Grid объект класса Column Checkbox;
2. Привязать сервис и корректно настроить атрибуты column, displayvalue, idproperty, valuefield;
3. Указать значение атрибута collectionvalues = array для передачи массива данных;
4. При необходимости добавить Filter Panel.

Поле "Выпадающая таблица-дерево" (Field Grid Tree)

Позволяет выбирать в поле значение из таблицы типа "Дерево".

Основные настройки

Важно!

Для корректной работы иерархии обязательно должно содержать объект класса [колонка "Дерево"](#)

Может содержать в себе:

- колонки;
- панель детализации;
- фильтр.

Сортировка данных в гриде настраивается с помощью атрибутов:

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

Для корректной работы поля необходимо указать сервис, возвращающий данные в сомбобоах 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	Наименование параметра, который будет отображен	ключ-значение, которое будет отображено в интерфейсе
idproperty	Наименование параметра, отвечающего за уникальность записей	ключ-значение, которое содержит уникальный идентификатор записи
valuefield	Наименование параметра из внутреннего сервиса, значение которого будет передано для дальнейшей обработки	ключ-значение из сервиса Field Grid Tree, из которого берется значение

Настройка размера выпадающего грида настраивается с помощью атрибутов:

Наименование атрибута	Описание
pickerheight	Высота выпадающей таблицы/списка. По умолчанию 390
pickerwidth	Ширина выпадающей таблицы/списка

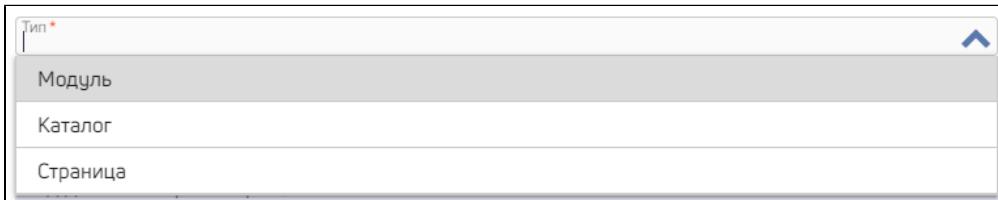
Особенности сервиса

Для корректного отображения иерархической структуры грида типа "Дерево" в результатирующем 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.

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

idproperty	Наименование параметра, отвечающего за уникальность записей	ключ-значение из сервиса combobox, содержащее уникальный ключ записи
displayfield	Наименование параметра, который будет отображен	ключ-значение из сервиса combobox, которое будет отображено в интерфейсе
valuefield	Наименование параметра из внутреннего сервиса, значение которого будет передано для дальнейшей обработки	ключ-значение из сервиса combobox, из которого берется значение

Пример

1. Сервис combobox возвращает выборку, где:

ck_id - уникальный идентификатор записи;
cv_name - наименование

2. После выбора одного значения из combobox, ИД выбранной записи должно быть передано на модификацию с наименованием параметра ck_parameter

3. Тогда настройка атрибутов Combobox будет выглядеть следующим образом:

Наименование атрибута	Значение	Комментарий
column	ck_parameter	ИД выбранной записи, передаваемой в Modify
idproperty	ck_id	ИД уникальной записи из списка в Combobox
displayfield	cv_name	Наименование из списка Combobox
valuefield	ck_id	ИД записи из списка в Combobox

В атрибуте allownew указывается префикс-метка для нового значения.

Пример: значение атрибута "allownew" = "NEW:".

При вводе нового значения в combobox на сервер уйдет значение "NEW:введенное значение"

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	Пауза (в сек) до вызова сервиса с применением фильтра	
querymode	Режим вызова сервиса: 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, по которому	по какому ключ-значению из сервиса

фильтруются значения при вводе

combobox производится фильтрация.

Пагинация значений combobox настраивается с помощью атрибута pagesize.

Для настройки значения по умолчанию используется атрибут **defaultvalue**.

Для выбора первого значения, возвращаемого сервисом, необходимо указать значение **##first##**.

Для выбора всегда первого значения необходимо указать значение **##alwaysfirst##**

Для создания небольших списков в Combobox непосредственно на фронте, без привязки к бэку необходимо:

1. создать в Combo столько дочерних объектов класса **Record**, сколько значений будет в справочнике. Например, для создания справочника их двух значений: Да и Нет, необходимо создать 2 дочерних объекта класса Record.
2. для каждого Record задаем в атрибутах valuefield и displayfield - ключ и значение соответственно. Например, valuefield =1, displayfield =Да
3. в Combo указываем значения в атрибутах: valuefield = valuefield, displayfield =displayfield, idproperty = valuefield

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)

Обратите внимание!

При указании расширения файла следует использовать MIME-type. Например: image/jpeg, image/svg+xml и т.д.

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 чтобы 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 создания объекта типа "Песочные часы".

The screenshot shows a dialog box titled 'Настройка благоустройства' (Setup for landscaping). It contains two main sections: 'Справочник благоустройств' (Landscaping catalog) on the left and 'Благоустройства на объекте обслуживания' (Landscaping on the service object) on the right. Both sections have a header with a back arrow icon and a sorting arrow icon. The left section lists items like 'Внутренняя система водоснабжения', 'Наружная система водоснабжения', etc. The right section lists items like 'Закрытая система водоснабжения', 'Индивидуальное отопление', etc. Between the two sections are arrows pointing from left to right and right to left. At the bottom right of the dialog is a button labeled 'ЗАКРЫТЬ' (Close).

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".

The screenshot shows a dialog box with two checkboxes. The top checkbox is checked and labeled 'Отображать в меню' (Display in menu). The bottom checkbox is unchecked and labeled 'Статическая страница' (Static page).

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:

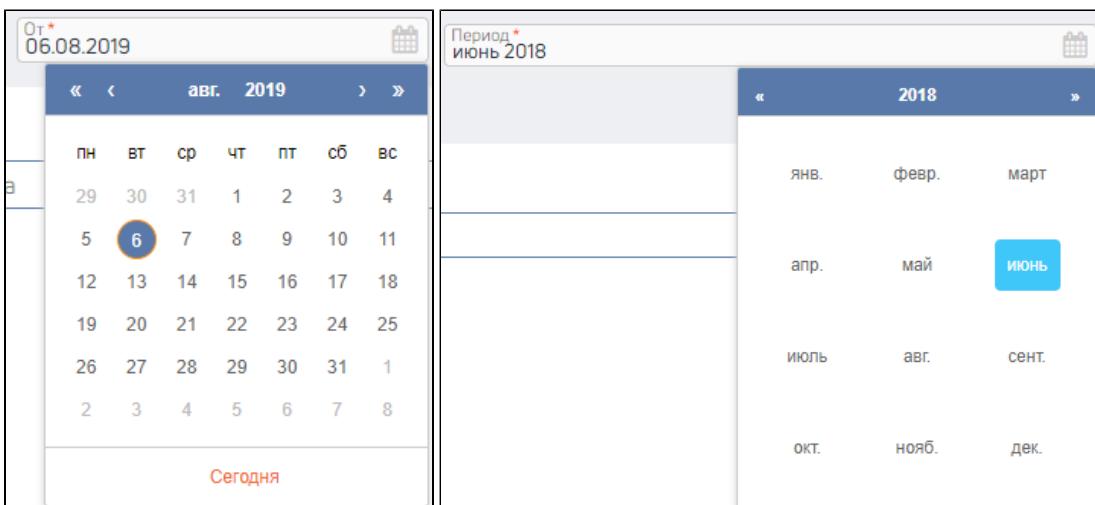
- 1 - ГГГГ;
- 2 - МММ ГГГГ;
- 3 - ДД.ММ.ГГГГ;
- 4 - ДД.ММ.ГГГГ ЧЧ:00;
- 5 - ДД.ММ.ГГГГ ЧЧ:МИ;
- 6 - ДД.ММ.ГГГГ ЧЧ:МИ:СС.

С помощью атрибутов minvalue / maxvalue можно задать минимальное / максимальное значение поля.

Возможно указание статичных значений в кавычках (например, "2019-12-20T00:00:00"), глобальных переменных либо синтаксиса условного оператора (условие ? выражение1 : выражение2)

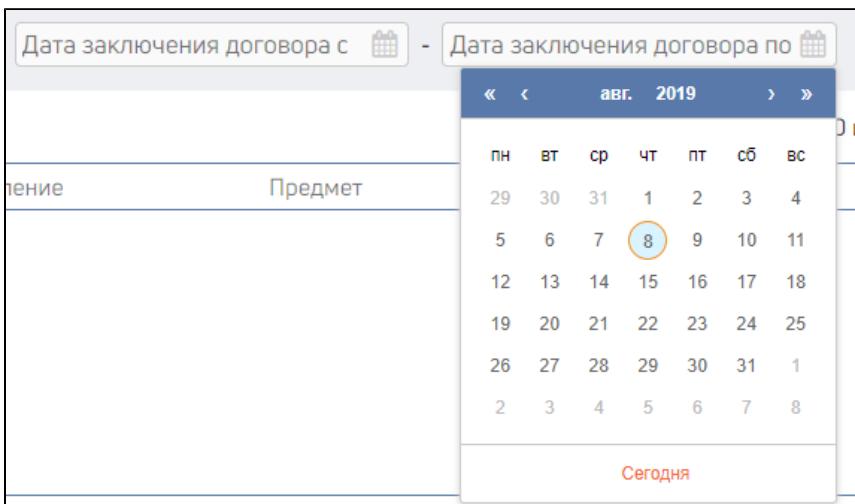
Поле "Дата" (Field Date)

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	Наименование параметра - начала периода
columnend	Наименование параметра - 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.

f_tarea

```
Буря мглою небо кроет,
Вихри снежные крутя;
То, как зверь, она завоет,
То заплачет, как дитя,
То по кровле обветшалой
Вдруг соломой зашумит,
То, как путник запоздалый,
К нам в окошко застучит.
```

Изменить высоту поля в режиме редактирования можно, потянув за правый нижний уголок поля

f_tarea

Буря мглою небо кроет,
Вихри снежные круты;
То, как зверь, она завоет,
То заплачет, как дитя,
То по кровле обветшалой
Вдруг соломой зашумит,
То, как путник запоздалый,
К нам в окошко застучит.

Атрибуты, позволяющие настроить высоту поля:

- minheight - высота, с которой отображаются поля при инициализации. Если данный атрибут задан, можно увеличить высоту поля, но сделать менее заданной высоты - нельзя.
- maxheight - максимальный предел высоты поля. При достижении данного предела включается скролл.

Поле "Пароль" (Field Password)

Компонент для ввода и хранения паролей и других конфиденциальных данных.

Пароль учетной записи БД *

5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8

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)

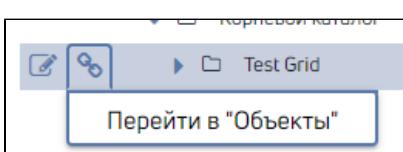
Скрытое поле (Field Hidden)

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 произведен 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.

Компоненты базового представления

В данном разделе представлена информация по компонентам, позволяющим настраивать базовое представление отображаемого интерфейса.

Формирование и настройка базового представление производится на странице "Приложения" (Меню CORE Приложения).

Указанная страница не имеет графического интерфейса и не отображается в Меню. Настройка производится путем добавления необходимых объектов на указанную страницу и переопределения их атрибутов.

The screenshot shows the 'Applications' page in the Core Application menu. At the top, there is a toolbar with icons for creating, deleting, and managing components. Below the toolbar, the title 'Наименование страницы' is displayed. A tree view shows the following structure:

- Корневой каталог
- CORE
 - Конструктор GUI
 - Администрирование
 - Авторизация
 - Локализация
 - УСПО
 - Автотесты
 - Приложения
 - Default Page
 - Меню

The 'Приложения' component is currently selected, indicated by a blue background.

Приложение (Application)

Основной элемент базового представления, в который добавляются дочерние элементы.

В атрибуте **activerules** указывается значение префикса приложения, при котором при формировании интерфейса будет использоваться данное приложение.

Пример

```
activerules = cv_url === "sample"
```

Выбранное приложение будет использовано при указании URL страницы вида: <http://адрес платформы/sample/id> страницы

Дочерними классами для Application являются:

1. Навигационная панель страницы (App Bar Panel)
2. Контейнер для контента страниц (Pages)
3. Окно (Window)
4. Окно "Выдвижная панель" (Window Drawer)

Контейнеры для контента страниц (Pages и Pager)

Элемент для размещения контента страниц.

При создании автоматически включает в себя объект класса Pager (Контейнер для контента страницы).

Для одного Application следует создавать только один контейнер Pages.

При создании нескольких Pages в одном Application контейнеры с контентом разместятся друг другом и будут отображать наполнение страницы. Однако, такая компоновка влияет на корректность работы контента.

The screenshot shows a user interface for managing application components. At the top, there are tabs for 'Страницы' (Pages), 'Объекты' (Objects), 'Атрибуты' (Attributes), and 'Классы' (Classes). Below the tabs, there are two main sections: 'Контейнеры страниц' (Page Containers) and 'Объекты' (Objects).

Контейнеры страниц (Page Containers):

Наименование страницы	Код действия просмотра	Код действия модификации	Порядок ↑
Core			1
Default Page			9
Меню			10
test			100

Объекты (Objects):

Наименование объекта	Наименование класса	Отображаемое имя	Описание	Порядок ↑	Мастер
Корневой каталог					

Below these sections, there are two more identical sets of tables for 'Контейнеры страниц' and 'Объекты', showing the same data as the first ones.

Контейнер Pager можно также размещать в Window Drawer. Тогда в раскрывающейся панели будет отображено наполнение страницы.

Подробнее [тут](#)

Наименование	Описание	Признак выводимого набор...	Признак самостоятельного...
App Bar Panel	Навигационная панель страницы	Нет	Да
Application	Контейнер для приложения	Да	Да
Badge Btn	Числовой значек над кнопкой	Нет	Нет
Box	Контейнер для объектов (облегч...	Нет	Нет
Button	Кнопка	Нет	Нет
Button Collector	Собиратель кнопок	Нет	Нет
Button Group	Контейнер кнопок	Нет	Да
Column Action Icon	Колонка-кнопка с иконкой	Нет	Да
Column Boolean	Колонка "Boolean" (да/нет)	Нет	Нет
Column Checkbox	Колонка "Флаг" (для мультивыб...	Нет	Нет
Column Date	Колонка "Дата"	Нет	Нет
Column Delete	Колонка-кнопка "Удаление"	Нет	Нет
Column Detail	Колонка-кнопка для отображен...	Нет	Нет

Атрибуты Дочерние классы Родительские классы

Тип атрибута: Все Основной Поведение Расположение Системный Отображение

+ Добавить

Наименование	Значение	Признак обязательн...	Описание	Тип атрибута

Навигационная панель страницы (App Bar Panel)

Панель для отображения:

1. кнопок (Button);
2. групп кнопок (Button Group);
3. собирателей кнопок (Button Collector);
4. вкладок открытых страниц (Open Page Tabs).

Группы кнопок (Button Group)

Класс, объединяющий в себе несколько кнопок (Button) для отображения в качестве единой группы в Навигационной панели (App Bar Panel).

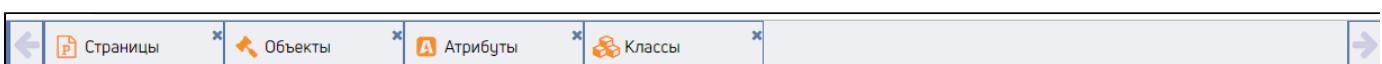
Порядок отображения кнопок соответствует порядку сортировки в Button Group.

Кнопки в группе будут разделены:

- вертикальной линией при contentview = hbox:
- 
-
- горизонтальной линией при contentview = vbox:
- 

Вкладки открытых страниц (Open Page Tabs)

Класс для отображения вкладок открытых страниц.



Окно "Выдвижная панель" (Window Drawer)

Стилизация класса окно (Window) под slide-panel.

Может включать в себя:

1. контейнеры (Box, Panel, Dynamic panel);
2. контейнер для отображения контента страницы (Pager);
3. список избранных страниц (Favorite Pages);
4. дерево страниц (Pages Tree);
5. панель оповещений (Notification Tab Panel);
6. выбор темы (Theme Combo).

Расположение окна задается атрибутом **align**.

Доступные значения:

- **left** - окно будет расположено с левой стороны экрана;
- **right** - окно отобразится справа.

Высота и ширина задается атрибутами **height** и **width** соответственно.

Атрибут **top** влияет на величину отступа от верхней границы экрана.

Окно Drawer вызывается кнопкой, расположенной в Навигационной панели (App Bar Panel).

Для корректной работы необходимо:

1. указать одинаковые значения атрибута **ckwindow** для Window Drawer и кнопки, которая его вызывает;
2. указать в качестве мастера для вызывающей кнопки - Приложение (Application);
3. указать для вызывающей кнопки значение атрибута **handler** = **onWindowOpen**.

Если в Window drawer помещен контейнер Pager, то в окне будет отображаться контент страницы, Идентификатор которой указан в атрибуте **defaultvalue** для Pager.

Пример:

The screenshot shows the constructor component library interface. At the top, there are tabs for 'Страницы' (Pages), 'Объекты' (Objects), 'Атрибуты' (Attributes), and 'Классы' (Classes). Below the tabs is a toolbar with icons for creating, deleting, and editing components.

Component List:

Наименование	Описание	Признак выводимого набор...	Признак самостоятельного...
App Bar Panel	Навигационная панель страницы	Нет	Да
Application	Контейнер для приложения	Да	Да
Badge Btn	Числовой значек над кнопкой	Нет	Нет
Box	Контейнер для объектов (облегч...	Нет	Нет
Button	Кнопка	Нет	Нет
Button Collector	Собиратель кнопок	Нет	Нет
Button Group	Контейнер кнопок	Нет	Да
Column Action Icon	Колонка-кнопка с иконкой	Нет	Да
Column Boolean	Колонка "Boolean" (да/нет)	Нет	Нет
Column Checkbox	Колонка "Флаг" (для мультивыб...	Нет	Нет
Column Date	Колонка "Дата"	Нет	Нет
Column Delete	Колонка-кнопка "Удаление"	Нет	Нет
Column Detail	Колонка-кнопка для отображен...	Нет	Нет

Attribute Configuration:

Below the component list is a section for attribute configuration. It includes tabs for 'Атрибуты' (Attributes), 'Дочерние классы' (Child Classes), and 'Родительские классы' (Parent Classes). A dropdown menu for 'Тип атрибута' (Attribute Type) is open, showing options: 'Все' (All), 'Основной' (Main), 'Поведение' (Behavior), 'Расположение' (Location), 'Системный' (System), and 'Отображение' (Display). There is also a 'Добавить' (Add) button and a toolbar with icons for creating, deleting, and editing attributes.

Наименование	Значение	Признак обязательн...	Описание	Тип атрибута

Работа с компонентами конструктора

- Redirect извне по сессии/токену
- Redirect извне с подстановкой фильтров
- Redirect на страницу
- Reusable objects (Переиспользуемые объекты)
- Глобальные переменные
 - Применение глобальных переменных
- Загрузка/выгрузка файлов по кнопке
 - Построчный разбор при загрузке файлов XLSX и CSV
- Как настроить объект "Песочные часы"

Redirect извне по сессии/токену

Назначение

Для интеграции с другими проектами реализован бесшовный переход на сайт. В данном случае авторизация не требуется, переход осуществляется с передачей сессии, либо токена (см. Формирование url для перехода).

Формирование страницы

Меню с выбором страниц всегда открыто, контент страниц располагается справа от меню. Шапка сайта с закладками, оповещением, выбором темы и информацией скрыта, тема берется по ранее выбранному варианту на общем сайте.

The screenshot shows the CORE application's interface. On the left, there is a sidebar with various menu items under 'CORE' such as 'Конструктор GUI', 'Страницы', 'Объекты', 'Классы', 'Атрибуты', 'Сервисы', 'Модули', 'Администрирование', 'Авторизация', 'УСПО', 'Отчеты', and 'Test'. The main area has two tabs: '+СОЗДАТЬ СТРАНИЦУ' and '+ДОБАВИТЬ ОБЪЕКТ'. The first tab displays a tree view of pages under 'Назначение страницы' with nodes like 'Корневой каталог', 'CORE', 'Меню', 'Отчеты', 'Test', and 'Где я?'. The second tab shows a table for adding objects with columns: 'Наименование объекта', 'Наименование класса', 'Описание', 'Порядок ↑', and 'Мастер'. There is also a 'Код действия просмотра' and 'Код действия модификации' section at the top of the main content area.

Формирование url для перехода

Переход на страницу осуществляется по одному из двух вариантов:

1. по сессии
 2. по токену
- В случае перехода по сессии необходимо сформировать url следующего вида:

<root-path-to-site>/frame/session/:session_id/:app/:page_id

где:

1. root-path-to-site - протокол передачи данных (http\https) + доменное имя сайта на который осуществляется переход,
2. frame/session - маршрут для перехода на страницу по сессии,
3. session_id - сессия пользователя для авторизации,
4. app - наименование [Приложения](#) для перехода,
5. page_id - Идентификатор страницы, на которую осуществляется переход

- В случае перехода по токену необходимо сформировать url следующего вида:

<root-path-to-site>/frame/token/:token_id/:app/:page_id

где:

1. root-path-to-site - протокол передачи данных (http\https) + доменное имя сайта на который осуществляется переход,
2. frame/token - маршрут для перехода на страницу по токену,
3. token_id - токен пользователя для авторизации,
4. app - наименование [Приложения](#) для перехода,
5. page_id - Идентификатор страницы, на которую осуществляется переход

Проблемы с сессией\токеном

При проблемах с сессией\токеном происходит разлогин, после можно залогинится. В момент логина происходит перезапись сессии

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: "eyJwYWdlIjoiMSIslmZpbHRlcil6eyjjdl9uYW1lIjoiQWNjb3VudCBHcmIkIn19"
4. Формируем ссылку для перехода: <http://адрес core/redirect/eyJwYWdlIjoiMSIslmZpbHRlcil6eyjjdl9uYW1lIjoiQWNjb3VudCBHcmIkIn19>

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
- Супер глобальные переменные

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 взят 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](#)

Супер глобальные переменные

Супер глобальные переменные - это переменные которые приходят из:

1. системных настроек, все что начинается с `g_sys`
2. тема, локализация - прописаны в системных настройках
3. сессия, начинается с `g_sess`

Эти переменные автоматически распространяются на все страницы и могут быть доступны в настройках атрибутов.

Их нельзя задавать на странице "Глобальные переменные на странице", т.к. это перебивает их значения и по сути является оверрайдом для этих переменных

Применение глобальных переменных

- Инициализация
- Атрибут setrecordtoglobal
- Атрибут Getglobal
 - Конкатенация ("склеивание") значений
- Атрибут Getglobaltostore
- Поведенческие (Behavior) атрибуты

Важно!

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

При необходимости можно задать значение глобальной переменной при инициализации страницы.

Для этого необходимо:

1. перейти на страницу Страницы (CORE Конструктор GUI Страницы);
2. выбрать страницу, для которой будет задаваться глобальная переменная;
3. нажать кнопку "Добавить", если создается новая глобальная переменная, или кнопку "Редактировать" для уже имеющейся глобалки;

Наименование объекта	Наименование класса	Отображаемое имя	Описание	Порядок ↑	Мастер
SYS Grid Class << DO NOT CHANGE	Grid	Таблица с классами		1	SYS Grid Class << DO NOT CHANGE
SYS Tab Panel Class << DO NOT CHANGE	Tab Panel	Панели со вкладками		2	

At the bottom, there are tabs for 'Атрибуты объекта на странице' and 'Глобальные переменные на странице'. A 'Добавить' (Add) button is visible.

4. в поле "Значение при инициализации" указать необходимое значение и нажать кнопку "Сохранить".

Наименование глобальной переменной	Значение при инициализации	Объект, задающий глобальную переменную	Объект, получающий глобальную переменную	Описание
Найменование глобальной переменной	Значение при инициализации			Описание

Атрибут setrecordtoglobal

При инициализации передает в глобальную переменную всю выбранную строку в виде объекта.

Пример: `g_test: {"ck_id": "all", "cv_name": "Все", "cv_description": "все атрибуты"}`

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
activerules	Правила для выбора активного элемента Используется для Field Combobox и Application.
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

('dept' 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

Исходный код
gv_name==1

Результат
false

Тип данных в результате: boolean

Переменные

Значение для gv_name
2

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
```

Построчный разбор при загрузке файлов XLSX и CSV

Для реализации необходимо:

1. создать плагин extractorfiletojson с корректными настройками

Пример настройки плагина

Редактирование плагина

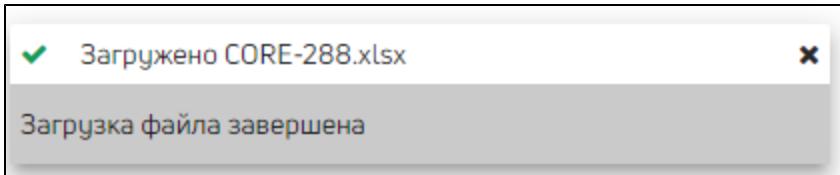
Наименование *	extractorfiletojson
Провайдер *	meta
Наименование плагина *	extractorfiletojson
Применять всегда	<input type="checkbox"/>
Порядок *	50
Применять если не указаны другие плагины	<input type="checkbox"/>
Описание	
dir aws riak	dir
Адрес Riak Dir Aws	/tmp
Наименование корзины S3	
Id key S3 Storage	
Secret key S3 Storage	
Количество строк при вызове	500000
Устанавливать права доступа public, добавляемым файлам в riak/aws	<input type="checkbox"/>
Разделитель колонок CSV	:
<input type="button" value="СОХРАНИТЬ"/> <input type="button" value="ОТМЕНА"/>	

2. создать сервис для сохранения разобранного файла. Например, TestXLSX
3. создать кнопку (Button) для загрузки файлов
4. у кнопки указываем значения атрибутов:

Наименование атрибута	Значение	Комментарий
extraplugingate	extractorfiletojson	Наименование созданного плагина разбора
mode	8	Режим загрузки файла
updatequery	TestXLSX	Наименование созданного сервиса разбора и хранения

Загрузка файла выглядит следующим образом:

1. Нажать кнопку, вызывающую загрузку
2. в появившемся окне выбрать файл для загрузки
3. при успешной загрузке система отобразит сообщение об успешном завершении операции



4. ответ от плагина загрузки будет иметь вид:

```
*{success: true, data: [{...}], metaData: {responseTime: 1.156, total: 1}}
  *data: {...}
    *0: {...}
      *data: {filter: [{}], jl_sort: [{direction: "ASC"}], jn_fetch: 1000, jn_offset: 0}, master: {...}
        ck_id: "auto-5ecc9088-e217-429d-bbb0-41d9f1837286"
        cv_file_guid: "41176572-1e71-4df2-a98f-ddd46144fdfa"
        cv_file_mime: "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet"
        cv_file_name: "CORE-288.xlsx"
      *extract_rows: [{A: "align", B: "Расположение текста:<br>left - слева,<br>center - по центру,<br>right - справа",...},...]
        *[0 ... 99]
          *0: {A: "align", B: "Расположение текста:<br>left - слева,<br>center - по центру,<br>right - справа",...}
          *1: {A: "allownew", B: "Признак доступности ввода нового значения (по умолчанию = false)", C: "behavior"}
          *2: {A: "autoload", B: "Признак автозагрузки сервиса", C: "basic"}
          *3: {A: "autoselectidentity",...}
          *4: {A: "bottombtn", B: "Служебный параметр для иерархии", C: "placement"}
          *5: {A: "btncollectorall", B: "Показывать иконку с аудитом для грида", C: "basic"}
          *6: {A: "btncollectorall", B: "Признак сбора статических кнопок в коллектор", C: "basic"}
          *7: {A: "btndelete", B: "Показывать иконку с удалением для грида", C: "basic"}
          *8: {A: "btnexcel", B: "Показывать иконку с загрузкой excel для грида", C: "basic"}
          *9: {A: "btncprint", B: "Показывать иконку для печати", C: "basic"}
          *10: {A: "btrefresh", B: "Показывать иконку с обновлением для грида", C: "basic"}
          *11: {A: "btncsettings", B: "Показывать кнопку настройками пользователя", C: "basic"}
          *12: {A: "charttype", B: "Тип графика lineChar | barChart | areachart | pieChart", C: "basic"}
          *13: {A: "checkaddmore", B: "Чекбокс для добавления еще одной записи", C: "behavior"}
          *14: {A: "childids", B: "Служебный параметр для иерархии", C: "placement"}
          *15: {A: "childwindow", B: "Служебный параметр для иерархии", C: "placement"}
          *16: {A: "ckwindow", B: "Идентификатор для поиска окна в мастере", C: "basic"}
          *17: {A: "collapsed",...}
          *18: {A: "collapsible", B: "Признак возможности сворачивания панели true/false", C: "behavior"}
```

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. к обоим гридам должны быть привязаны сервисы;
 3. модификатор сервиса должен уметь работать с массивом данных;
 4. указываем значение атрибута selmode = MULTI на обоих гридах;
 5. чтобы спрятать управляющие кнопки на гридах, указываем значение атрибута hideactions = true на обоих гридах.

Настройка мастеров

Пример корректной настройки мастеров при заведении дополнительных характеристик сущности с помощью объекта типа "Песочные часы":

1. Главный грид "Роли" (Roles Grid) - список всех ролей в системе;

2. Подчиненный грид "Действия" (Role Actions Grid)- список действий, которые используются в указанной роли;

The screenshot shows a grid-based interface for managing roles and their actions. At the top, there is a search bar and a toolbar with buttons for adding, deleting, and saving. Below this, the main area is divided into two sections: 'Наименование' (Name) and 'Описание' (Description). The 'Наименование' section lists several roles: 'asdaddfsf', 'test21', 'test_auth', 'zbr_17', 'Администратор', 'Консультант', and 'zbr_17'. The 'Описание' section provides details for each role. The 'Администратор' role is highlighted. Below this grid, there is another section titled 'Действия' (Actions) and 'Кто использует' (Who uses it), which contains a table of actions with their descriptions.

Наименование	Описание
asdaddfsf	
test21	test
test_auth	тестирование авторизации - без конструктора
zbr_17	x
Администратор	Доступ ко всем страницам на просмотр и редактирование
Консультант	Доступ ко всем страницам на просмотр

Наименование	Описание
491 Доступ на просмотр страницы "Атрибуты"	Доступ на просмотр страницы "Атрибуты"
492 Доступ на модификацию страницы "Атрибуты"	Доступ на модификацию страницы "Атрибуты"
497 Доступ на просмотр страницы "Классы"	Доступ на просмотр страницы "Классы"
498 Доступ на модификацию страницы "Классы"	Доступ на модификацию страницы "Классы"
499 Доступ на просмотр страницы "Объекты"	Доступ на просмотр страницы "Объекты"
500 Доступ на модификацию страницы "Объекты"	Доступ на модификацию страницы "Объекты"
503 Доступ на просмотр страницы "Профиль"	Доступ на просмотр страницы "Профиль"

3. Окно "Настроить действия роли" в гриде "Действия" - "Песочные часы" (Field Item Selector) с 2-я гридами:

1. "Доступные действия" (Available Action Grid);
2. "Выбранные действия" (Selected Action Grid)

The screenshot shows the 'Настройте действия роли' (Configure Role Actions) dialog. It features two side-by-side grids. The left grid, titled 'Доступные действия' (Available Actions), lists actions with their codes and descriptions. The right grid, titled 'Выбранные действия' (Selected Actions), lists actions that have been selected for a specific role. A central area between the grids contains arrows for moving items between them. At the bottom right of the dialog is a 'ЗАКРЫТЬ' (Close) button.

Код действия...	Наименование	Описание
100 Тестовый код	Тестовый код	Тестовый код
110 Тестовый код 22	Тестовый код 24	Тестовый код 24
120 Тестовый код 3	Тестовый код 3	Тестовый код 3

Код действия...	Наименование	Описание
491 Доступ на просмотр страницы "Атрибуты"	Доступ на просмотр страницы "Атрибуты"	Доступ на просмотр страницы "Атрибуты"
492 Доступ на модификацию страницы "Атрибуты"	Доступ на модификацию страницы "Атрибуты"	Доступ на модификацию страницы "Атрибуты"
497 Доступ на просмотр страницы "Классы"	Доступ на просмотр страницы "Классы"	Доступ на просмотр страницы "Классы"
498 Доступ на модификацию страницы "Классы"	Доступ на модификацию страницы "Классы"	Доступ на модификацию страницы "Классы"
499 Доступ на просмотр страницы "Объекты"	Доступ на просмотр страницы "Объекты"	Доступ на просмотр страницы "Объекты"
500 Доступ на модификацию страницы "Объекты"	Доступ на модификацию страницы "Объекты"	Доступ на модификацию страницы "Объекты"
503 Доступ на просмотр страницы "Профиль"	Доступ на просмотр страницы "Профиль"	Доступ на просмотр страницы "Профиль"
504 Доступ на модификацию страницы "Профиль"	Доступ на модификацию страницы "Профиль"	Доступ на модификацию страницы "Профиль"

4. Для корректной работы необходимо указать грид "Роли" в качестве мастера у следующих объектов:

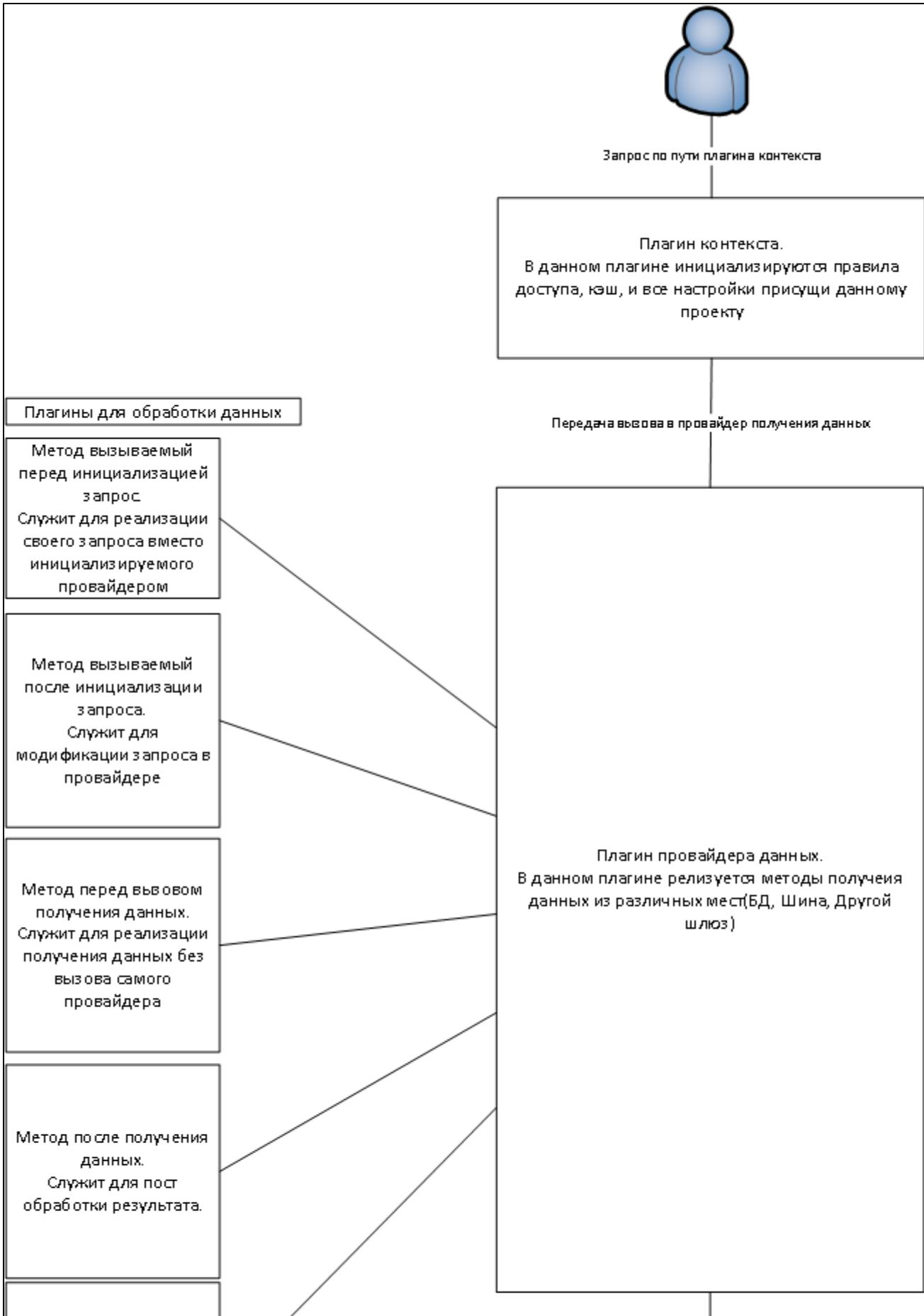
1. грид "Действия" (Role Actions Grid);
2. объект "Песочные часы" (Field Item Selector);
3. грид "Доступные действия" (Available Action Grid);
4. грид "Выбранные действия" (Selected Action Grid);

The screenshot shows the Object Explorer (левая панель) and the Properties panel (правая панель). The Object Explorer displays a tree structure of objects, including 'Roles Grid << DO NOT CHANGE', 'Roles Tabs << DO NOT CHANGE', 'Role actions Grid', 'Settings button', 'Settings Window', 'Field Item Selector', 'Available Action Grid', 'Selected Action Grid', and 'Close Button'. The Properties panel shows various properties for the 'Role actions Grid' object, such as 'Справочник ролей' (Role Grid), 'Roles Tabs', 'Список действий роли' (List of actions for the role), 'Настройте действия роли' (Configure Role Actions), 'Field Item Selector', 'Доступные действия' (Available Actions), 'Выбранные действия' (Selected Actions), and 'Закрыть' (Close).

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

- 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

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

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
nsiGateUrl	Ссылка на гейт NSI

AuthMock

Плагин для реализации заглушек 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
<code>core</code>	режим с инициализацией темповых таблиц
<code>user</code>	Наименование учетной записи БД
<code>password</code>	Пароль учетной записи БД
<code>connectString</code>	Строка подключения к БД
<code>poolMin</code>	Минимальное количество коннектов к БД в пуле
<code>poolMax</code>	Максимальное количество коннектов к БД в пуле

PostgreSQLDb

Плагин вызова запросов в БД Postgres.

▼ [Настройки](#)

Наименование параметра	Описание
<code>core</code>	Инициализация по схеме CORE
<code>connectString</code>	Строка подключения к БД
<code>poolMax</code>	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
<code>columns</code>	Наименования колонок, которые надо распаковать (через запятую)
<code>extractSingleColumn</code>	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
prefixPath	Префикс пути для формирование ссылки на скачивание
queryNotification	Наименование запроса для оповещения
maxExcelRows	Максимальное количество строк при печати в Excel
exportUrl	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 оповещения

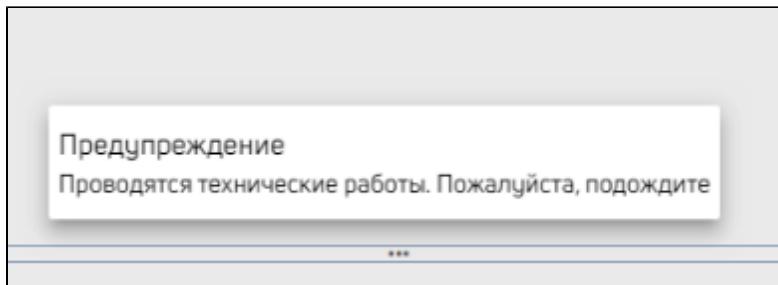
В платформе CORE реализована система уведомлений, которая включает в себя три блока:

- семафоры;
- оповещения из БД Postgres;
- оповещения из БД Oracle.

Семафоры - сообщение о недоступности системы (например, при сбросе кэша).

Анализируется таблица t_semaphore. Если cn_value <> 0, то отображается сообщение о технических работах.

Данный тип сообщений отображается для всех авторизованных пользователей.



Оповещения из БД (Oracle/Postres)- сообщения об ошибках и предупреждениях системы.

Анализируется таблица t_notification. При добавлении записи в t_notification БД оповещает шлюз.

В указанную таблицу сохраняются данные о дате возникновения ошибки/предупреждения, ИД пользователя, получившего ошибку, признаке отправки и данные о сообщении(код ошибки из t_message, макросы динамического текста, ИД страницы и ИД объекта на странице, вернувшего ошибку). Данный вид оповещений будет отображаться персонифицированно: только тому пользователю, кто вызвал ошибочную операцию.

Пример

cv_message

```
{"cv_error":{"47":["1_1"]}, "reloadpageobject":{"ck_page":"541810","ck_page_object":"7BE1EFFCAA306BBDE053809BA8COAD87"}}
```



Тип сообщения: ошибка (error), предупреждение (warning) либо информационное сообщение (info) - определяется для каждого кода ошибки в таблице t_message (cr_type).

Все сообщения/ошибки/предупреждения отображаются в [Панели оповещения](#) в соответствии с их типом.

Настройки системы уведомлений указываются в конфигурационном файле t_events.toml.

Подробнее об обработке ошибок [тут](#)

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
- Загрузка модуля (zip)

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 состояние 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 информации 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 туре 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 страницы, 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.

Загрузка модуля (zip)

Для разворачивания модуля на стенде необходимо:

1. собрать с помощью команды yarn build
2. создать архив для передачи yarn zip

После создания архив будет доступен в папке dist.

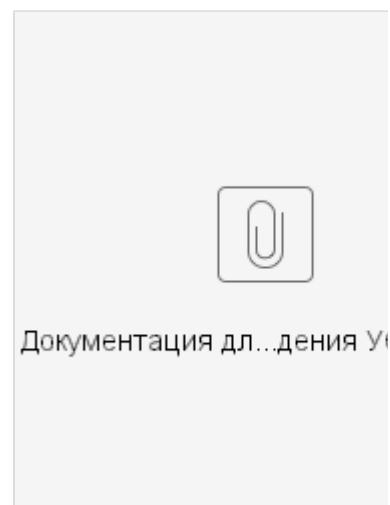
Имя zip файла создается по шаблону \${имя-модуля}-\${версия-модуля}.zip

Модуль содержит:

1. исходные файлы самого модуля, которые располагаются в dist
2. файл манифеста, который хранится по пути проекта src/schema_manifest.json

Модуль УСПО

- 1. Введение
 - 1.1 Цель
 - 1.2 Определения и сокращения
- 2. Требования к квалификации системного администратора
- 3. Назначение и условия применения
 - 3.1 Назначение
 - 3.2 Условия применения
- 4. Описание операций
 - 4.1 Установка s3 хранилища
 - 4.1.1 Установка
 - 4.2 Развёртывание базы данных
 - 4.2.1 Развёртывание базы данных
 - 4.3 Установка Wildfly
 - 4.3.1 Установка Wildfly
 - 4.3.2 Подключение модуля Postgres JDBC
 - 4.3.3 Подключение датасорса
 - 4.4 Развёртывание УСПО
 - 4.4.1 Установка web и receiver
 - 4.4.2 Установка service
 - 4.4.3 Подключение к CORE



1. Введение

1.1 Цель

Данный документ содержит последовательность действий по разворачиванию «Универсальной системы печати отчетности».

1.2 Определения и сокращения

Термин	Описание
--------	----------

УСПО	Универсальная система печати отчетности
ПО	Программное обеспечение
ИР	Информационные ресурсы
АПК	Аппаратно-программный комплекс

2. Требования к квалификации системного администратора

Для обслуживания Системы администратор должен обладать следующими навыками:

- Навыки работы в серверных Linux либо опыт администрирования *nix подобных систем;
- Опыт администрирования БД PostgreSQL 10+;

Для поддержки Системы системному администратору необходимо:

- Ознакомиться с документацией производителей используемого аппаратного и системного программного обеспечения;

3. Назначение и условия применения

3.1 Назначение

«Универсальная система печати отчетности» представляет собой комплексное решение, для формирования отчетов

3.2 Условия применения

Для работы Системы на сервере предварительно следует установить:

Сервер	ПО
Сервер БД УСПО	СУБД PostgreSQL 10+
	ОС CentOS 7+
Сервер хранения отчетов	ОС CentOS 7+
Сервер приложений	ОС CentOS 7+
	OpenJDK 11+

4. Описание операций

4.1 Установка s3 хранилища

riak cs — свободная реализация s3 хранилища

riak kv — свободная nosql БД

4.1.1 Установка

1. Добавляем репозиторий riak /etc/yum.repos.d/basho.repo

```
[basho_riak]
name=basho_riak
baseurl=https://packagecloud.io/basho/riak/el/7/$basearch
```

```

repo_gpgcheck=1
gpgcheck=0
enabled=1
gpgkey=https://packagecloud.io/basho/riak/gpgkey
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
metadata_expire=300

[basho_riak-cs]
name=basho_riak-cs
baseurl=https://packagecloud.io/basho/riak-cs/el/7/$basearch
repo_gpgcheck=1
gpgcheck=0
enabled=1
gpgkey=https://packagecloud.io/basho/riak-cs/gpgkey
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
metadata_expire=300

[basho_stanchion]
name=basho_stanchion
baseurl=https://packagecloud.io/basho/stanchion/el/7/$basearch
repo_gpgcheck=1
gpgcheck=0
enabled=1
gpgkey=https://packagecloud.io/basho/stanchion/gpgkey
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
metadata_expire=300

```

2. Установка Riak KV

▼ Вызвать установку

```
yum install riak
```

3. Установка Riak CS

▼ Вызвать установку

```
yum install riak-cs
```

4. Установка Stanchion

▼ Вызвать установку

```
yum install stanchion
```

5. Настраиваем /etc/riak/advanced.config

```

[{"lager,
 [
  {extra_sinks,
   [
    {object_lager_event,
     [{handlers,
       [{"lager_file_backend,
         [{"file, "/var/log/riak/object.log"}],
         {level, info},
         {log_level, info}
       }
     ]
   }
 ]
}
]
```

```

        {formatter_config, [date, " ", time, " [",
severity, " ] ", message, "\n"]}}
    ]
}
},
{async_threshold, 500},
{async_threshold_window, 50}]
}
]
}
],
{eleveldb, [
{total_leveldb_mem_percent, 30}
]},
{riak_kv, [
%% Other configs
{add_paths, ["/usr/lib64/riak-cs/lib/riak_cs-2.1.1/ebin"]},
%% -- riak cs
{storage_backend, riak_kv_multi_backend},
{multi_backend_prefix_list, [{<<"0b:>>, be_blocks}]}},
{multi_backend_default, be_default},
{multi_backend, [
{be_default, riak_kv_eleveldb_backend, [
{data_root, "/var/lib/riak/leveldb"}]
},
{be_blocks, riak_kv_bitcask_backend, [
{data_root, "/var/lib/riak/bitcask"}]
]}
]}
%% Other configs
]},
{riak_core, [
%% Other configs
{default_bucket_props, [{allow_mult, true}]}},
%% Other configs
]}].

```

6. Включаем авто запуск Riak KV

▼ Показать команду
`systemctl enable riak`

7. Стартуем Riak KV

▼ Показать команду
`systemctl start riak`

8. Включаем авто запуск Stanchion

▼ Показать команду
`systemctl enable stanchion`
`systemctl start stanchion`

9. Стартуем Stanchion

▼ Показать команду
`systemctl start stanchion`

10. Настраиваем Riak CS /etc/riak-cs/riak-cs.conf

```
listener = 0.0.0.0:10011
anonymous_user_creation = on
admin.listener = 127.0.0.1:1101
```

11. Включаем авто запуск Riak CS

▼ Показать команду
systemctl enable riak-cs

12. Стартуем Riak CS

▼ Показать команду
systemctl start riak-cs

13. Создаем пользователей и сохраняем доступы

```
Admin
curl -H 'Content-Type: application/json' -XPOST http://127.0.0.1:11011/riak-cs/user --data '{"email":"admin@example.com", "name": "admin"}'
{"email":"admin@example.com", "display_name": "admin", "name": "admin", "key_id": "LQPKAD2AUQIMU_RWF9KH", "key_secret": "RLBTEONm_zr8nrAmQSge-kEt5P5qsKM9boYLPw==", "id": "be2aeacc3c4532d7c245e3d4192c5705321d4620ef903b275b95d0a1d1662d27", "status": "enabled"}
Report
curl -H 'Content-Type: application/json' -XPOST http://127.0.0.1:11011/riak-cs/user --data '{"email": "report@example.com", "name": "report"}'
{"email": "report@example.com", "display_name": "report", "name": "report", "key_id": "V7CBOR8N--CYOCX_FQOU", "key_secret": "RIEzm3o3myteoLNW_10aH5NVq_mXQOwF19hWuQ==", "id": "604ae623a335f8cfa92972d15f09f746f9a585ddbc82d325a8b3f714e6138080", "status": "enabled"}
```

14. Перенастраиваем /etc/riak-cs/riak-cs.conf

```
anonymous_user_creation = off
#admin.listener = 127.0.0.1:1101
admin.key = LQPKAD2AUQIMU_RWF9KH
admin.secret = RLBTEONm_zr8nrAmQSge-kEt5P5qsKM9boYLPw==
```

15. Перенастраиваем /etc/stanchion/stanchion.conf

```
admin.key = LQPKAD2AUQIMU_RWF9KH
admin.secret = RLBTEONm_zr8nrAmQSge-kEt5P5qsKM9boYLPw==
```

16. Перезапускаем Stanchion

▼ Показать команду
systemctl restart stanchion

17. Перезапускаем Riak CS

▼ Показать команду

```
systemctl restart riak-cs
```

4.2 Развёртывание базы данных

4.2.1 Развёртывание базы данных

Для развёртывания базы данных Системы на сервере БД УСПО необходимо установить СУБД PostgreSQL.

1. Создать пользователей:

Суперпользователь, от которого идет заливка бд:

```
CREATE ROLE s_su WITH
    LOGIN
    SUPERUSER
    INHERIT
    CREATEDB
    CREATEROLE
    NOREPLICATION;
```

Пользователь для входа шлюза:

```
CREATE ROLE s_uc WITH
    LOGIN
    NOSUPERUSER
    INHERIT
    NOCREATEDB
    NOCREATEROLE
    NOREPLICATION;

ALTER ROLE s_uc SET search_path TO uspo, public, pg_catalog;
```

Пользователь без возможности авторизации, которому будут принадлежать функции и процедуры:

```
CREATE ROLE s_up WITH
    NOLOGIN
    NOSUPERUSER
    INHERIT
    NOCREATEDB
    NOCREATEROLE
    NOREPLICATION;
```

Пользователь для входа java:

```
CREATE ROLE uspo WITH
    LOGIN
    NOSUPERUSER
```

```
INHERIT  
NOCREATEDB  
NOCREATEROLE  
NOREPLICATION;  
  
ALTER ROLE uspo SET search_path TO uspo, public, pg_catalog;
```

2. Создать контейнер БД:

```
CREATE DATABASE uspo  
WITH  
OWNER = s_su  
TEMPLATE = template0  
ENCODING = 'UTF8'  
LC_COLLATE = 'ru_RU.UTF-8'  
LC_CTYPE = 'ru_RU.UTF-8'  
TABLESPACE = pg_default  
CONNECTION LIMIT = -1;
```

3. Распаковать архив и настроить liquibase.uspo.properties

```
driver: org.postgresql.Driver  
url: jdbc:postgresql://127.0.0.1:5432/uspo  
username: s_su  
password: s_su
```

4. Настроить liquibase.meta.properties — должна смотреть на БД core.

5. Запустить update.

4.3 Установка Wildfly

4.3.1 Установка Wildfly

Установка производится согласно инструкции https://docs.wildfly.org/17/Getting_Started_Guide.html

4.3.2 Подключение модуля Postgres JDBC

Скачиваем <https://jdbc.postgresql.org/download.html> и распаковываем, например, /opt/pjdbc

Подключаем к wildfly

```
#cd /opt/wildfly-17.0.0.Final/#.bin/jboss-cli.sh  
module add --name=org.postgresql --slot=main --resources=/opt/pjdbc  
/postgresql-jdbc.jar --dependencies=javax.api,javax.transaction.api  
connect  
/subsystem=datasources/jdbc-driver=postgres:add(driver-name="postgres",  
driver-module-name="org.postgresql",driver-class-name=org.postgresql.  
Driver)
```

4.3.3 Подключение датасурса

1. Заходим в администрирование wildfly

The screenshot shows the JBoss Management Console interface. The top navigation bar has tabs: Homepage, Deployments, Configuration (which is selected), Runtime, Patching, and Access Control. Under Configuration, there's a 'Configuration' section with links to Subsystems, Interfaces, Socket Bindings, Paths, and System Properties. The 'Subsystems' link is expanded, showing Batch, JBoss EJB, Core Management, Datasources & Drivers (which is selected and highlighted in blue), Deployment Scanners, Discovery, Distributable Web, and EE. To the right, under 'Datasources & Drivers', there's a 'Datasources' section with 'JDBC Drivers'. A list of existing datasources shows 'ExampleDS' and 'uspo'. On the far right, there's a 'Datasource' section with a 'Filter by: name, xa, .../disabled' input field and a large 'Add' button, which is also highlighted with a red box.

2. Создаем датасурс uspo, JNDI: java:jboss/datasources/print_server

The screenshot shows the 'Add Datasource' wizard. The title bar says 'Add Datasource'. Below it is a horizontal navigation bar with six steps: 'Choose Template' (step 1, highlighted in blue), 'Attributes' (step 2), 'JDBC Driver' (step 3), 'Connection' (step 4), 'Test Connection' (step 5), and 'Review' (step 6). The main area is titled 'Attributes'. It contains two fields: 'Name' with the value 'uspo' and 'JNDI Name' with the value 'java:jboss/datasources/print_server'. A note at the bottom says 'Required fields are marked with *'. The entire 'Attributes' section is highlighted with a red box. At the bottom right are buttons for 'Cancel', '< Back', and 'Next >'.

4.4 Развёртывание УСПО

УСПО состоит из:

- web – приложение которое формирует отчет (war приложение)
- receiver – приложение прием запросов добавление в очередь (war приложение)
- service – приложение обработки очереди (jar приложение)

4.4.1 Установка web и receiver

Заходим в администрирование

The screenshot shows the HAL Management Console interface. The top navigation bar includes 'Homepage', 'Deployments' (which is selected), 'Configuration', 'Runtime', and 'Patching'. Below this, the main area displays 'Deployment (2)'. Two items are listed: 'uspo-print-receiver_prom_rev_f5e70...' and 'uspo-print-web_prom_rev_0f0a409.war'. A context menu is open over the first item, with options: 'Upload Deployment' (highlighted in blue), 'Add Unmanaged Deployment', and 'Create Empty Deployment'. A tooltip on the right side of the menu provides instructions: 'Drag and drop to add new deployment'.

4.4.2 Установка service

1. Распаковываем, например: `/opt/uspo/service`
Настраиваем `/opt/uspo/service/uspo_print_service.properties`

```
db.url=jdbc:postgresql://127.0.0.1:5432/uspo
db.user=uspo
db.pass=uspo
uspo.web.endpoint=http://127.0.0.1:8000/uspo-print-web
/print -- web
```

2. Настраиваем `/opt/uspo/service/daemon.sh`

```
#!/bin/sh
export PATH=$JAVA_HOME/bin:$PATH
DIRNAMESTART=`dirname "$0"`
PROGNAMESTART=`basename "$0"`
DIR="$(dirname $(readlink -f $0))"
```

```

cd $DIR
pid=`pgrep -f "java.*-jar $DIR"`
start() {
if [ ! -z $pid ]; then
    echo "USPO Service is already running pid $pid"
    exit 1
fi
nohup java -Dlog4j.configuration=file:$DIR/log4j.properties -jar
$DIR/uspo-print-service.jar $DIR/uspo_print_service.properties
2>&1 </dev/null | cat >> $DIR/logs/service.log &
}

stop() {
if [ ! -z $pid ] ; then
kill $pid
else
    echo "USPO Service do not running"
    exit 1
fi
}

case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    restart)
        stop
        start
        ;;
    *)
        echo "Usage: $PROGNAMESTART start|stop|restart"
        exit 1
esac

exit 0

```

3. Настраиваем запуск

[Показать команду](#)

```
chmod +x /opt/uspo/service/daemon.sh
```

4. Настраиваем сервис /etc/systemd/system/uspo-service.service

```

[Unit]
Description=USPO Service
Documentation=USPO Service
After=network.target

[Service]
Type=forking

```

```
#User=uspo-wildfly
LimitNOFILE=infinity
LimitNPROC=infinity
LimitCORE=infinity

ExecStart=/opt/uspo/service/daemon.sh start
ExecReload=/opt/uspo/service/daemon.sh restart
ExecStop=/opt/uspo/service/daemon.sh stop

[Install]
WantedBy=multi-user.target
```

5. Включаем в авто старт

▼ Показать команду

```
systemctl enable uspo-service
```

6. Подключаемся к бд иср, правим настройки

```
-- 
UPDATE uspo.t_d_global
SET cv_value='127.0.0.1:10011'
WHERE ck_id='cv_riak_url';
--   riak cs
UPDATE uspo.t_d_global
SET cv_value='reports'
WHERE ck_id='cv_riak_bucket';--id key  report
UPDATE uspo.t_d_global
SET cv_value='V7CBOR8N--CYOCX_FQ0U'
WHERE ck_id='cv_riak_access_key';
--secret report
UPDATE uspo.t_d_global
SET cv_value='RIEzm3o3myteoLNW_10aH5NVq_mXQOWF19hWuQ=='
WHERE ck_id='cv_riak_secret_access_key';
--   web
UPDATE uspo.t_d_global
SET cv_value='http://127.0.0.1:7080/uspo-print-web/print'
WHERE ck_id='print_servlet_url';
--   core
UPDATE uspo.t_d_global
SET cv_value='http://127.0.0.1:8080/api?
action=dml&query=MTAddNotification&plugin=PrintJasperServer'
WHERE ck_id='external_rest_url';
--   ,
UPDATE uspo.t_d_global
SET cv_value='/opt/vdir'
WHERE ck_id='jr_virtualizer_directory_name';
```

7. Стартуем сервис

▼ Показать команду

```
systemctl restart uspo-service
```

8. Перезапускаем приложения в wildfly

4.4.3 Подключение к CORE

1. Добавляем провайдер ..t_providers.toml

```
[ [data]]
ck_id = "uspo"
cl_autoload = 0
ck_d_plugin = "postgresqldb"
cv_description = "    "

[ data.cct_params ]
connectString = "postgres://s_uc:s_uc@127.0.0.1:5432/uspo"
poolMax = 50
```

2. Добавляем плагин ..t_plugins.toml

```
[ [data]]
cv_name = "printjasperserver"
ck_d_provider = "all"
ck_d_plugin = "USPOIntegration"
cv_description = "    "
cl_required = 0
cl_default = 0
cn_order = 3

[ data.cct_params ]
callBackUrl = http://127.0.0.1:8080
/api
    urlReceiver = http://127.0.0.1:7080/uspo-print-
receiver
    -- receiver
    prefixPath = "/reportserver"
    exportUrl = "c821e316-173d-4f71-b6a5-6a16f3bde6b3"
    queryNotification = "MTAddNotification"
```

Быстрый старт

Раздел с описанием последовательности действий для создания и настройки компонентов Платформы.

Шаг 0. Создание и настройка провайдера

Для корректного взаимодействия Платформы с пользовательской БД, в которой хранятся бизнес-данные, необходимо создать провайдера, который будет обращаться к данной БД.

Для этого необходимо:

1. переходим на вкладку "Настройка провайдеров" на странице "Шлюз" (Меню CORE Администрирование Шлюз):

Идентификатор	Наименование	Описание	Авторизация
admingate	Шлюз	Провайдер администрирования	Администрирование
auth	Шлюз	Провайдер авторизации	Авторизация
authcore	Метамодель	Провайдер для тестовых данных	
core_test	Провайдер для смородины		
meta	УСЛО		
smor_test			
uspo			

2. Добавляем нового провайдера БД: указываем Уникальный идентификатор и наименование нового провайдера и нажимаем кнопку "Сохранить":

Идентификатор*	Наименование*
admingate	Шлюз
auth	Авторизация

3. На той же вкладке добавляем провайдера шлюза с тем же именем:

Наименование	Признак загрузки при инициализации	Наименование плагина	Описание	Настройки
admingate	Нет	admingate	Провайдер администрирования	cvRiaUrl=http://172.16.211.60:1001...
auth	Да	authmock	Провайдер авторизации	adminUser=admin_core...
authcore	Да	coreauthpg	Провайдер внутренней авторизации	connectString=***..

4. Заполняем поля формы и нажимаем кнопку "Сохранить":

Добавление провайдера	
Наименование *	<input type="text"/>
Признак загрузки при инициализации	<input type="checkbox"/>
Наименование плагина *	<input type="text"/>
Описание	<input type="text"/>
<input type="button" value="СОХРАНИТЬ"/> <input type="button" value="ОТМЕНА"/>	

Доступные для заполнения поля:

Наименование поля	Тип данных	Комментарии
Наименование	Текст	Обязательно для заполнения
Признак загрузки при инициализации	Да/Нет	Загружать ли при старте шлюза
Наименование плагина	Выбор из списка	Обязательно для заполнения

Описание

Текст

Далее привязываем плагин обработки запроса к созданному провайдеру.

1. переходим на вкладку "Плагины" на странице "Шлюз" (Меню CORE Администрирование Шлюз) и нажимаем кнопку "Добавить":

Наименование	Наименование провайдера	Наименование плагина	Обязательный плагин	По умолчанию	Порядок	Описание	Настройки
extractrow	authcore	jsonrowcolumnextractor	Да	Нет	1	columns=json.result..	
extractrow	meta	jsonrowcolumnextractor	Да	Нет	2	Плагин распаковывания вложенног..	extractSingleColumn=true..
extractrow	core_test	jsonrowcolumnextractor	Да	Нет	5	extractSingleColumn=true..	extractSingleColumn=true..
extractrow	uspo	jsonrowcolumnextractor	Да	Нет	2	Плагин распаковывания вложенног..	extractSingleColumn=true..
modulestorage	meta	modulestorage	Нет	Нет	1	Место хранения моделей	cvtTypeStorage=riak..
preparequery	uspo	preparequery	Да	Нет	1	Плагин преобразования запросов..	

2. Заполняем поля формы и нажимаем кнопку "Сохранить":

Наименование *
Провайдер *
Наименование плагина *
Применять всегда
Порядок *
Применять, если не указаны другие плагины
Описание
СОХРАНИТЬ ОТМЕНА

Доступные для заполнения поля:

Наименование поля	Тип данных	Комментарии
Наименование	Текст	Обязательно для заполнения
Провайдер	Выбор из списка	Обязательно для заполнения
Наименование плагина	Выбор из списка	Обязательно для заполнения
Применять всегда	Да/Нет	
Порядок	Число	Обязательно для заполнения В каком порядке запускать плагины провайдера
Применять, если не указаны другие плагины	Да/Нет	
Описание	Текст	

3. Для корректной модификации запросов, применения фильтров и сортировки (макросы &SORT, &FILTER) необходимо привязать плагин - **preparequery**

4. Для корректной распаковки ответа из определенных колонок строки в json необходимо привязать плагин - **jsonrowcolumn extractor**

Шаг 1. Грид на отображение

Для простого старта работы с Платформой разберем пример: создадим страницу с таблицей, в которой будет содержаться информация о списке абонентов.

- БД
- Сервис
- Объекты и их настройки
- Страница
- Объект на странице
- Итог

БД

Создадим в БД таблицу t_abonent с данными:

CK_ID	CV_SURNAME	CV_NAME	CN_PHONE
1	Иванов	Иван	5663398
2	Петров	Петр	5669833
3	Сидоров	Сидор	5663893

ck_id - Уникальный идентификатор записи

cv_surname - Фамилия

cv_name - Имя

cn_phone - номер телефона

Обратите внимание!

В соответствии со [Стандартом разработки](#) в каждой таблице должен присутствовать первичный ключ, который именуется ck_id

Сервис

Создадим сервис ShowAbonent для выборки всех данных из таблицы t_abonent.

```
select ck_id, cv_surname, cv_name, cn_phone from t_abonent
```

Обратите внимание!

В соответствии со [Стандартом разработки](#) каждый сервис должен содержать в себе параметр ck_id для определения уникальности записи

Созданный сервис добавляем в таблицу s_mt.t_query.

Объекты и их настройки

Создадим объект класса Грид для отображения данных абонентов:

1. переходим на страницу "Объекты" (Меню CORE Конструктор GUI Объекты);

Наименование	Описание	Отображаемое имя	Имя сервиса	Класс	Используется на страницах	Метод модификации	Порядок сортировки
Корневой каталог							1
Wysiwyg panel	Wysiwyg panel			Panel	WYSIWYG		1
YCTO: Test_USPO_Report	Тестирование отчета УСПО		USPOShowReportPage	Grid	Test USPO Report	pkg	1
ABRR - NDS GRID	Сумма НДС			Grid	Справочники		10
ABRR - Conformity Grid	Соответствия			Grid	Справочники		10
ABRR - Subservice Unit	Единицы измерения		SMGetMeasureTypeForSubService	Grid	Виды платежей		10

2. Выбираем корневой каталог, нажимаем кнопку "Создать", заполняем поля и нажимаем кнопку "Сохранить":

Добавление Объекта

Класс*
Grid (Грид для отображения данных)

Сервис
ShowAbonent

Имя объекта*
Abonent Grid

Описание объекта*
Список абонентов

Порядок сортировки*
10

Отображаемое имя
Список абонентов

Метод модификации

Провайдер данных при модификации

СОХРАНИТЬ **ОТМЕНА**

3. Настраиваем атрибуты грида:

Наименование	Значение по умолчанию	Значение (переопределено)	Описание
defaultvalue			Значение по умолчанию.
edittype	inline		Режим добавления/редактирования: inline - в строке; modalwindow - в модальном окне.
extraplugins			Дополнительные плагины для шаблона.
height			Статическая высота в пикселях (px).
novalidate			Признак отключения глобального лаудера при загрузке сервиса.
orderdirection			Направление сортировки: ASC / DESC.
orderproperty			Имя колонки из сервиса, по которой сортируем.
Редактировать значение	false		Количество выводимых строк (включает пагинатор).
refreshgridrecords	false		Признак перезагрузки всех данных после сохранения/обновления.
reloadmaster			Признак обновления родителя по мастеру.
request	true		Признак зависимости от настера (true/false).
selmode	SINGLE		Возможность выделения нескольких значений в GRID и TREEGRID.

Наименование атрибута	Значение	Комментарий
orderproperty	ck_id	Наименование параметра, отвечающего за уникальность записи

4. Создадим колонку для отображения Фамилии наших абонентов:

1. В списке объектов выбираем созданный нами грид Abonent Grid, нажимаем кнопку "Создать" и заполняем поля:

Класс *
Column String (Колонка "Строка")

Сервис

Имя объекта *
cv_surname

Описание объекта *
Фамилия

Порядок сортировки *
10

Отображаемое имя
Фамилия

Метод модификации

Провайдер данных при модификации

СОХРАНИТЬ **ОТМЕНА**

2. Заполним атрибуты колонки:

Наименование атрибута	Значение	Комментарий
column	cv_surname	Наименование параметра из сервиса, откуда брать значения

5. Аналогично создаем колонки для Имени и Номера телефона

Рекомендую

При создании объектов задавать порядок сортировки кратный десяти: 10, 20, 30 и т.д.

Т.к. порядковый номер объекта должен быть уникальным в рамках родительского объекта, то такая нумерация позволяет при необходимости легко добавить еще 9 объектов между ранее созданными объектами.

6. Получаем Грид с тремя колонками:

Abonent Grid	Список абонентов	Список абонентов	ShowAbonent	Grid	Абоненты	10
cv_surname	Фамилия	Фамилия		Column String	Абоненты	10
cv_name	Имя	Имя		Column String	Абоненты	20
cv_phone	Номер телефона	Номер телефона		Column String	Абоненты	30

Страница

Создадим страницу для отображения нашего грида.

1. переходим на страницу «Страницы» (Меню CORE Конструктор GUI Страницы);

2. выбираем родительский модуль/каталог либо корневой каталог, в котором будет создана страница. В нашем примере выбираем Тестовый каталог и нажимаем кнопку «Создать»:

Наименование страницы	Код действия просмотра	Код действия модификации	Порядок ↑
Корневой каталог			1
Core			10
Меню			20
Отчеты			40
Test			1
Тестовая категория (не удалять)			
Test Grid	515	516	0
test page	515	700	1

3. Заполняем поля и нажимаем кнопку "Сохранить":

Добавление Страницы

Наименование *	Абоненты
Тип *	Страница
Порядок *	10
Иконка	fa-address-card-o
Код действия просмотра ?	515
Код действия модификации ?	516
Отображать в меню	<input checked="" type="checkbox"/>
Статическая страница	<input type="checkbox"/>
Ссылка ?	
<input type="button" value="СОХРАНИТЬ"/> <input type="button" value="ОТМЕНА"/>	

Объект на странице

Добавим наш Abonent Grid на созданную страницу "Абоненты".

1. Выбираем созданную страницу "Абоненты" и нажимаем кнопку "Добавить объект":

Список объектов

Наименование страницы	Код действия просмотра	Код действия модификации	Порядок ↑
Test	20	40	
Тестовая категория (не удалять)			1
Test Grid	515	516	0
test page	515	700	1
test order	516	516	10
test Romanyuk	515	516	10
Абоненты	515	516	10
еееее	511	512	58
zbr_test_page	515	515	77
zbr_mods	515	516	80

+ Добавить

Наименование объекта Наименование класса Описание Порядок ↑ Мастер

Корневой каталог

2. В списке объектов находим Abonent Grid и добавляем его на страницу:

Добавление объекта на страницу

Объект* Abonent Grid	<input type="button" value="▼"/>
Порядок* 10	
Выбор мастера	<input type="button" value="🔍"/>
<input type="button" value="СОХРАНИТЬ"/> <input type="button" value="ОТМЕНА"/>	

3. После добавление грида на страницу необходимо сбросить кеш, чтобы Платформа подгрузила в кеш вновь созданную страницу с объектами на ней:

Список объектов

Наименование страницы	Код действия просмотра	Код действия модификации	Порядок ↑
Test	20	40	
Тестовая категория (не удалять)			1
Test Grid	515	516	0
test page	515	700	1
test order	516	516	10
test Romanyuk	515	516	10
Абоненты	515	516	10
еееее	511	512	58
zbr_test_page	515	515	77
zbr_mods	515	516	80

+ Добавить

Наименование объекта Наименование класса Описание Порядок ↑ Мастер

Корневой каталог

Abonent Grid Grid Список абонентов 10

4. При сбросе кеша появится сообщение:

Предупреждение
Проводятся технические работы. Пожалуйста, подождите

5. Не пугаемся и ждем



Итог

После сброса кеша открываем меню, находим страницу "Абоненты" и открываем ее.

The screenshot shows a software interface with a sidebar menu. The 'Test' category is selected, revealing its contents: 'Test Grid' and 'test page'. The 'Абоненты' item is highlighted with an orange bar at the bottom of the sidebar. At the very bottom, there are two tabs: 'test order' and 'Абоненты', with 'Абоненты' being the active tab.

Любуемся и хвалим себя.

Фамилия	Имя	Номер телефона
Иванов	Иван	5663398
Петров	Петр	5669833
Сидоров	Сидор	5663893
...		

Шаг 2. Добавление/редактирование/удаление данных в гриде

Настроим, созданный в Шаге 1, грид для возможности добавления/редактирования/удаления данных.

БД

Необходимо создать функцию, описывающую добавление/редактирование/удаление данных из таблицы t_abonent.

Напоминаем:

Для корректной модификации данных в результате наборе данных обязательно должен присутствовать уникальный идентификатор (ck_id) модифицированной записи.

Наименование созданного модификатора указываем в поле "Метод модификации" для созданного Abonent Grid на странице Объекты. Так же выбираем провайдера данных

Редактирование Объекта

Класс* Grid (Грид для отображения данных)
Сервис ShowAbonent
Имя объекта* Abonent Grid
Описание объекта* Список абонентов
Порядок сортировки* 10
Отображаемое имя Список абонентов
Метод модификации pkg_json_abonent.add_abonent
Провайдер данных при модификации* Провайдер для тестовых данных

СОХРАНИТЬ **ОТМЕНА**

Объекты и их настройки

Создадим кнопки добавление и редактирования.

1. переходим на страницу "Объекты" (Меню CORE Конструктор GUI Объекты);
2. Создаем кнопку "Добавить". Выбираем Abonent Grid, нажимаем кнопку "Создать", заполняем поля и нажимаем кнопку "Сохранить":

Добавление Объекта

Класс *	Button (Кнопка)	▼
Сервис		▼
Имя объекта *	Add Button	
Описание объекта *	Добавить	
Порядок сортировки *	40	
Отображаемое имя	Добавить	
Метод модификации		
Провайдер данных при модификации		▼
		СОХРАНИТЬ
		ОТМЕНА

3. Настраиваем атрибуты кнопки добавления:

Наименование атрибута	Значение	Комментарий
handler	onCreateChildWindowMaster	Для вызова окна при создании
iconfont	plus	Наименование иконки из шрифта Font Awesome
mode	1	Добавление

4. Создаем колонку-кнопку "Редактировать". Выбираем Abonent Grid, нажимаем кнопку "Создать", заполняем поля и нажимаем кнопку "Сохранить":

Добавление Объекта

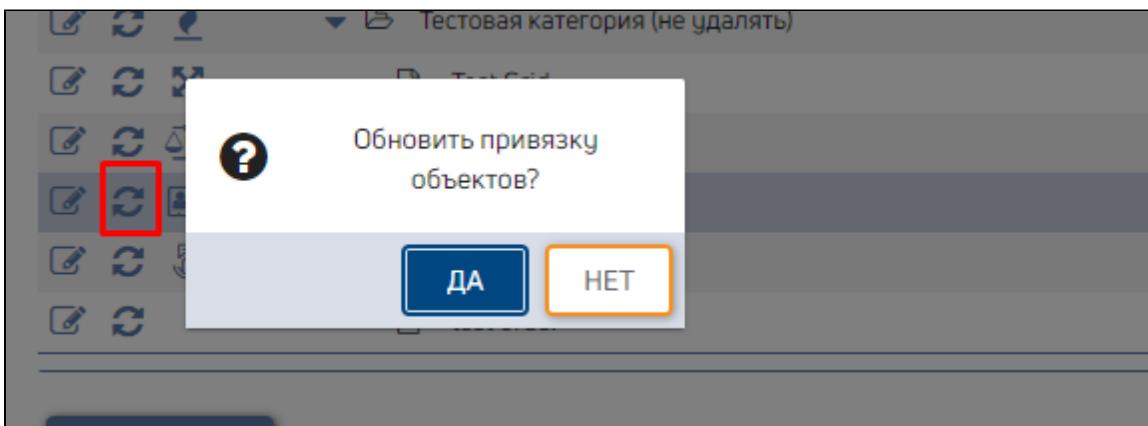
Класс*	Column Edit (Колонка-кнопка "Редактирование")	▼
Сервис		▼
Имя объекта*	Edit Button	
Описание объекта*	Редактировать	
Порядок сортировки*	50	
Отображаемое имя	Редактировать	
Метод модификации		
Провайдер данных при модификации		▼
		СОХРАНИТЬ
		ОТМЕНА

5. Атрибуты колонки-кнопки "Редактировать" уже настроены на классе. При желании их можно переопределить на объекте.

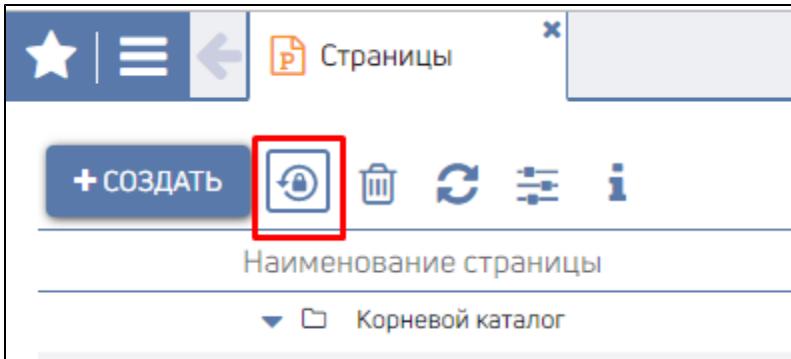
Объект на странице

После добавления кнопок на объект необходимо, чтобы они отобразились на странице. Для этого:

1. переходим на страницу "Страницы" (Меню CORE Конструктор GUI Страницы);
2. обновляем привязку объектов на странице "Абоненты":



3. после сбрасываем кеш на сервере:



4. Любимся на наши успехи:

Список абонентов		
Фамилия	Имя	Номер телефона
Иванов	Иван	5663398
Петров	Петр	5669833
Сидоров	Сидор	5663893
...		

Внутренняя авторизация

В Платформе реализована внутренняя авторизация через провайдера authcore, который смотрит на схему core_auth.

Администрирование авторизации осуществляется в блоке Авторизация (CORE Авторизация).

Обратите внимание!

Для немедленного применения модифицированных данных авторизации (Пользователи, роли, действия и т.д.) необходимо выполнить сброс провайдера авторизации.

Действия

Каждой странице Платформы соответствует пара действий:

- действие на просмотр страницы (только отображение страницы и навигация по ней; все кнопки с действиями (например, "Добавить" или "Редактировать" - заблокированы), доступен рефреш и пагинация, применение фильтров);
- действие на модификацию страницы (возможность модифицировать данные на странице).

Обратите внимание!

Если для страницы не указано действие на просмотр, то при попытке открыть указанную страницу Система вернет ошибку "Страница недоступна (нет прав доступа или страница не существует)"

Подробнее о создании и настройке действий

Роли

Роль представляет собой совокупность действий.

После добавления всех необходимых действий роль назначается Пользователям.

[Подробнее о создании и настройке ролей](#)

Пользователи

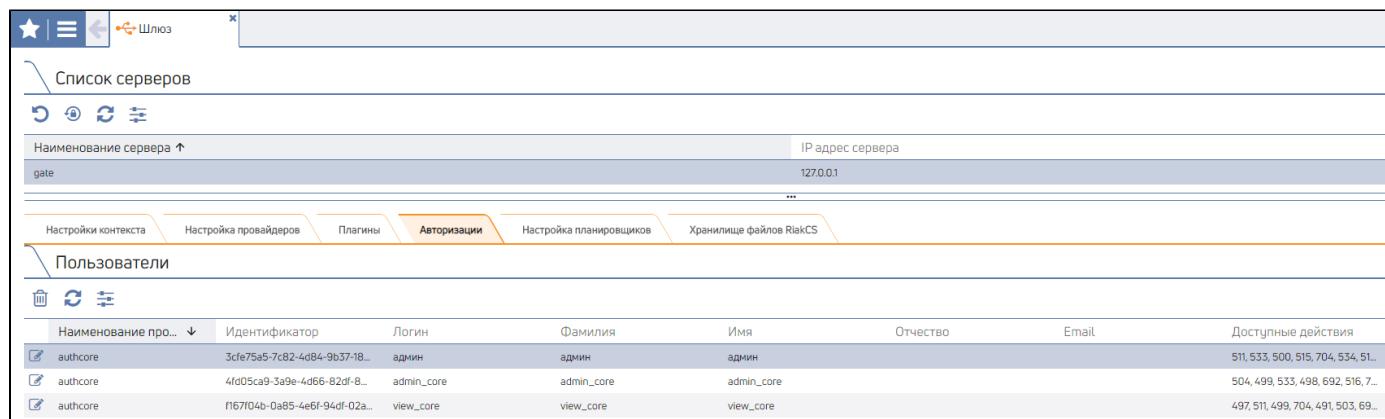
Пользователи также создаются в блоке Внутренней авторизации.

Пользователям назначаются Роли.

При необходимости можно вести дополнительную информацию по каждому пользователю.

[Подробнее о создании и настройке пользователей](#)

Список пользователей, их данные и доступные им действия можно посмотреть во вкладке "Авторизация" на странице "Шлюз" (Меню -> CORE -> Администрирование -> Шлюз):



The screenshot shows the 'Shluz' configuration interface with the 'Authorization' tab selected. The main table displays three users with their details and available actions:

Наименование про...	Идентификатор	Логин	Фамилия	Имя	Отчество	Email	Доступные действия
authcore	3cfe75a5-7c82-4d84-9b37-18...	админ	админ	админ			511, 533, 500, 515, 704, 534, 51...
authcore	4fd05ca9-3a9e-4d66-82df-8...	admin_core	admin_core	admin_core			504, 499, 533, 498, 692, 516, 7...
authcore	f167f04b-Da85-4e6f-94df-02a...	view_core	view_core	view_core			497, 511, 499, 704, 491, 503, 69...

Настройка действий

Обратите внимание!

Для немедленного применения модифицированных данных авторизации (Пользователи, роли, действия и т.д.) необходимо выполнить сброс провайдера авторизации.

Создание действия

Для создания действия необходимо:

1. перейти на страницу «Действия» (Меню -> CORE -> Авторизация -> Действия);
2. нажать кнопку «Добавить»;

Код действия

Наименование

+ Добавить

Код действия	Наименование
100	Тестовый код
110	Тестовый код 22
120	Тестовый код 3
491	Доступ на просмотр страницы "Атрибуты"
492	Доступ на модификацию страницы "Атрибуты"
497	Доступ на просмотр страницы "Классы"
498	Доступ на модификацию страницы "Классы"
499	Доступ на просмотр страницы "Объекты"
500	Доступ на модификацию страницы "Объекты"
503	Доступ на просмотр страницы "Профиль"

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

Добавление		
Код действия	Наименование	Описание
100	Тестовый код	Тестовый код

Доступные для заполнения поля:

Наименование поля	Тип данных	Комментарии
Код действия	Целое число	Если не указано значение, то при сохранении присваивается значение по сиквенсу.
Наименование	Текст	Обязательно для заполнения.
Описание	Текст	

Обратите внимание!

При попытке создать действия с одинаковым значением поля "Код действия" Система вернет ошибку и не выполнит сохранение.

Редактирование действия

Для редактирования действия необходимо:

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

Обратите внимание!

Поле "Код действия" отредактировать нельзя

Удаление действия

Для удаления действия необходимо:

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

Обратите внимание!

При попытке удалить действие, которое используется на какой-либо роли, Система вернет ошибку "Удаление невозможно, т.к. существуют связанные записи"

Настройка ролей

Обратите внимание!

Для немедленного применения модифицированных данных авторизации (Пользователи, роли, действия и т.д.) необходимо выполнить сброс провайдера авторизации.

Создание роли

Для создания роли необходимо:

1. перейти на страницу «Роли» (Меню -> CORE -> Авторизация -> Роли);
2. нажать кнопку «Добавить»;

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

Доступные для заполнения поля:

Наименование поля	Тип данных	Комментарии
Наименование	Текст	Обязательно для заполнения.
Описание	Текст	

Обратите внимание!

При попытке создать роль с одинаковым значением поля "Наименование" Система вернет ошибку и не выполнит сохранение.

Редактирование роли

Для редактирования роли необходимо:

1. выбрать роль;
2. нажать кнопку-колонку «Редактировать значение»;
3. внести изменения;

- нажать кнопку «Сохранить».

Наименование	Описание
asdaddfsf	Тестовый редактор
r_watcher	Тестовый наблюдатель

Удаление роли

Для удаления роли необходимо:

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

Добавление и настройка действий роли

После создания роли можно приступить к настройке списка кодов действия для нее.

Для этого необходимо:

- выбрать роль;
- в гриде "Действия" нажать кнопку "Настроить";

Наименование	Описание
asdaddfsf	Тестовый редактор
r_redactor	Тестовый наблюдатель
r_watcher	Роль для доступа к страницам Смородины
smor_test	тестирование авторизации - без конструктора
test_auth	Доступ ко всем страницам на просмотр и редактирования
Администратор	Доступ ко всем страницам на просмотр
Консультант	

Действия	Кто использует
НАСТРОЙКА	

- в открывшемся окне с левой стороны будут отображаться все доступные действия, в правой - выбранные.
Перемещение действий производится с помощью одинарных ("Добавить/удалить запись") и двойных ("Добавить/удалить всё");

Настройте действия роли

Доступные действия

Код действия...	Наименование	Описание
110	r_view-code	Тестовый код просмотра
120	r_mod-code	Тестовый код модификации
491	Доступ на просмотр страницы "Атрибуты"	Доступ на просмотр страницы "Атрибуты"
492	Доступ на модификацию страницы "Атрибуты"	Доступ на модификацию страницы "Атрибуты"
497	Доступ на просмотр страницы "Классы"	Доступ на просмотр страницы "Классы"
498	Доступ на модификацию страницы "Классы"	Доступ на модификацию страницы "Классы"
499	Доступ на просмотр страницы "Объекты"	Доступ на просмотр страницы "Объекты"
500	Доступ на модификацию страницы "Объекты"	Доступ на модификацию страницы "Объекты"
503	Доступ на просмотр страницы "Профиль"	Доступ на просмотр страницы "Профиль"
504	Доступ на модификацию страницы "Профиль"	Доступ на модификацию страницы "Профиль"
511	Доступ на просмотр страницы "Страницы"	Доступ на просмотр страницы "Страницы"
512	Доступ на модификацию страницы "Страницы"	Доступ на модификацию страницы "Страницы"
	...	

Выбранные действия

Код действия...	Наименование	Описание

>> > < <<

ЗАКРЫТЬ

4. при необходимости поиска определенного действия можно воспользоваться колоночным фильтром;

Настройте действия роли

Доступные действия

Код действия...	Наименование	Описание
110	Код действия	Тестовый код просмотра
491	Доступ на просмотр страницы "Атрибуты"	Доступ на просмотр страницы "Атрибуты"
492	Доступ на модификацию страницы "Атрибуты"	Доступ на модификацию страницы "Атрибуты"
497	Доступ на просмотр страницы "Классы"	Доступ на просмотр страницы "Классы"
498	Доступ на модификацию страницы "Классы"	Доступ на модификацию страницы "Классы"
499	Доступ на просмотр страницы "Объекты"	Доступ на просмотр страницы "Объекты"
500	Доступ на модификацию страницы "Объекты"	Доступ на модификацию страницы "Объекты"
504	Доступ на модификацию страницы "Профиль"	Доступ на модификацию страницы "Профиль"
511	Доступ на просмотр страницы "Страницы"	Доступ на просмотр страницы "Страницы"

<< < > >>

5. нажать кнопку "Закрыть". Все выбранные действия отобразятся в гриде "Действия".

The screenshot shows two grids. The top grid is titled 'Наименование' (Name) and lists roles: asdaddfsf, r_redactor, r_watcher, smor_test, and test_auth. The bottom grid is titled 'Код действия' (Action Code) and lists actions: 110_r_view-code and 120_r_mod-code. Both grids have columns for 'Наименование' (Name), 'Описание' (Description), and '...'.

Добавление и настройка пользователей, использующих роль

1. выбрать роль;
2. в гриде "Кто использует" нажать кнопку "Настроить";

The screenshot shows the 'Кто использует' (Who uses it) grid. It lists roles: asdaddfsf, r_redactor, r_watcher, smor_test, and test_auth. The 'Настроить' (Configure) button is highlighted with a red box. The grid has columns for 'Логин' (Login), 'Фамилия' (Surname), 'Имя' (Name), and 'Отчество' (Middle Name).

3. в открывшемся окне с левой стороны будут отображаться все доступные пользователи, в правой - выбранные. Перемещение пользователей производится с помощью одинарных ("Добавить/удалить запись") и двойных ("Добавить/удалить всё");

The screenshot shows the 'Настройте действия роли' (Configure role actions) dialog. It contains two tables: 'Доступные пользователи' (Available users) and 'Выбранные пользователи' (Selected users). The 'Доступные пользователи' table lists users: admin_core, ru_redactor, ru_watcher, smor_test, test1, test_auth, and view_core. The 'Выбранные пользователи' table currently has no entries. Between the tables are move buttons: '>', '<', and '>>'.

4. нажать кнопку "Закрыть". Все выбранные пользователи отобразятся в гриде "Кто использует".

The screenshot shows a table titled 'Кто использует' (Who uses) with the following columns: 'Наименование' (Name), 'Описание' (Description), 'Логин' (Login), 'Фамилия' (Last name), 'Имя' (First name), 'Отчество' (Middle name), and 'E-mail'. The table contains several rows, including 'test1' with 'test' in all other fields.

Наименование	Описание	Логин	Фамилия	Имя	Отчество	E-mail
asdaddsf	Тестовый редактор					
r_redactor	Тестовый наблюдатель					
r_watcher	Роль для доступа к страницам Смородины					
smor_test	тестирование авторизации - без конструктора					
test_auth						
...						

Настройка пользователей

Обратите внимание!

Для немедленного применения модифицированных данных авторизации (Пользователи, роли, действия и т.д.) необходимо выполнить сброс провайдера авторизации.

Добавление пользователя

Для создания пользователя необходимо:

1. перейти на страницу «Пользователи» (Меню -> CORE -> Авторизация -> Пользователи);
2. нажать кнопку «Добавить»;

The screenshot shows the 'Пользователи' (Users) page with a search bar and a table. A red box highlights the 'Добавить' (Add) button in the top toolbar. The table has columns: 'Логин' (Login), 'Фамилия' (Last name), and 'Имя' (First name). It contains two rows: 'admin_core' and 'view_core'.

Логин	Фамилия	Имя
admin_core		admin_core
view_core		view_core

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

Добавление Пользователя

Логин *

Пароль *

Фамилия *

Имя *

Отчество

E-mail

СОХРАНИТЬ

ОТМЕНА

Доступные для заполнения поля:

Наименование поля	Тип данных	Комментарии
Логин	Текст	Обязательно для заполнения
Пароль	Текст	Обязательно для заполнения
Фамилия	Текст	Обязательно для заполнения
Имя	Текст	Обязательно для заполнения
Отчество	Текст	
E-mail	Текст	

Обратите внимание!

При попытке создать пользователей с одинаковым значением поля "Логин" Система вернет ошибку и не выполнит сохранение.

Обратите внимание!

Если на странице "Список дополнительных данных пользователя" заданы какие-то значения, то они будут отображены в окне добавления Пользователя. [Подробнее](#)

Редактирование пользователя

Для редактирования пользователя необходимо:

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

Логин **админ**

Пароль

Фамилия **админ**

Имя **админ**

Отчество

E-mail

СОХРАНИТЬ **ОТМЕНА**

Обратите внимание!

Для обеспечения информационной безопасности в окне редактирование не отображается сохраненный ранее пароль пользователя.

Удаление пользователя

Для удаления пользователя необходимо:

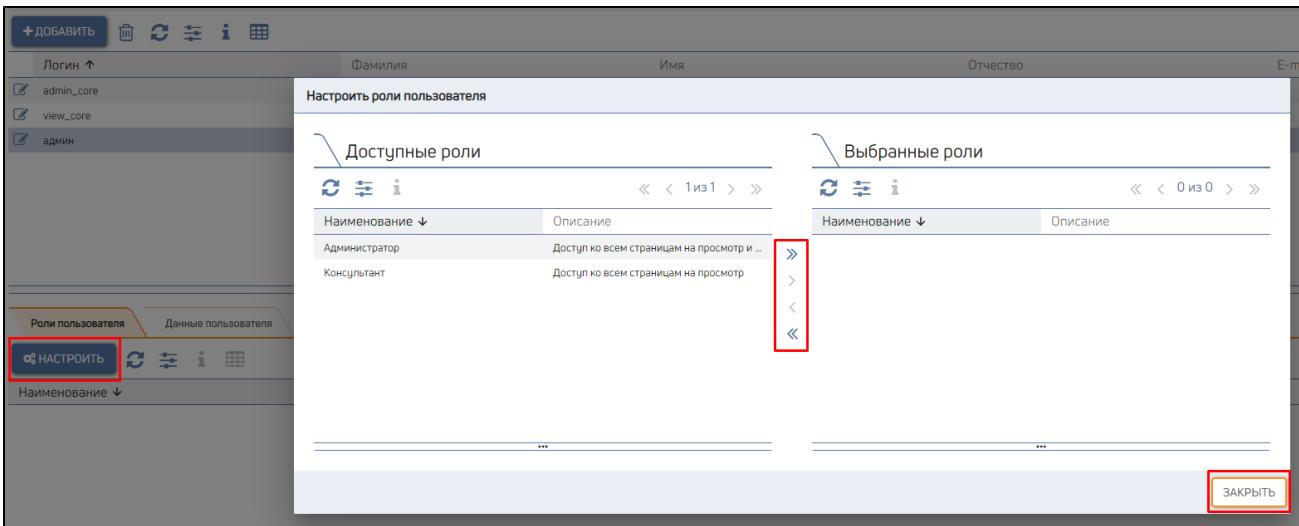
1. выбрать пользователя;
2. нажать "Удалить";
3. подтвердить удаление, нажав на кнопку «Да».

Удалить?

ДА **НЕТ**

Добавление и настройка ролей пользователя

1. Выбрать пользователя, для которого будет производится настройка ролей;
2. в гриде "Роли пользователя" нажать кнопку "Настроить";



3. в открывшемся окне с левой стороны будут отображаться все доступные роли, в правой - выбранные. Перемещение ролей производится с помощью одинарных ("Добавить/удалить запись") и двойных ("Добавить/удалить всё");
4. при необходимости поиска определенной роли можно воспользоваться колоночными фильтрами;

5. нажать кнопку "Закрыть".
- Все выбранные действия отобразятся в гриде "Роли пользователя".

Добавление и настройка дополнительных данных пользователя

Описание добавления данных в Справочник дополнительных данных пользователя тут

Для ранее созданных пользователей новая запись из Справочника будет отображаться в гриде "Данные пользователя" с пустым значением:

The screenshot shows a table titled 'Данные пользователя' (Additional Data User) with columns: Логин ↑ (Login ↑), Фамилия (Last Name), Имя (Name), Отчество (Middle Name), and E-mail. There are three rows: 'admin_core' (Фамилия: admin_core, Имя: admin_core, Отчество: empty, E-mail: empty), 'view_core' (Фамилия: view_core, Имя: view_core, Отчество: empty, E-mail: empty), and 'Админ' (Фамилия: Админ, Имя: Админ, Отчество: empty, E-mail: empty). Below the table, there are tabs for 'Роли пользователя' (User Roles) and 'Данные пользователя' (Additional Data User), with 'Данные пользователя' being active. A modal window at the bottom left shows fields for 'Описание' (Description) and 'Значение' (Value), with 'День рождения' (Birthday) selected.

Заполнить указанную информацию можно через редактирование записи:

The screenshot shows the same table and tabs as the previous screenshot. A modal window titled 'Редактирование Данных пользователя' (Edit Additional Data User) is open over the 'Admin' row. It contains a field 'День рождения*' (Birthday*) with a calendar icon. At the bottom right of the modal are 'СОХРАНИТЬ' (Save) and 'ОТМЕНА' (Cancel) buttons. A red box highlights the 'Edit' icon in the list table next to the 'Admin' row. A callout box at the bottom left says 'Обратите внимание!' (Attention!) with the note: 'При попытке удалить данные пользователя с Признаком обязательного заполнения = "Да" Система вернет ошибку и не произведет удаление.' (When trying to delete data for a user with the mandatory field indicator = "Yes", the system will return an error and not perform the deletion.)

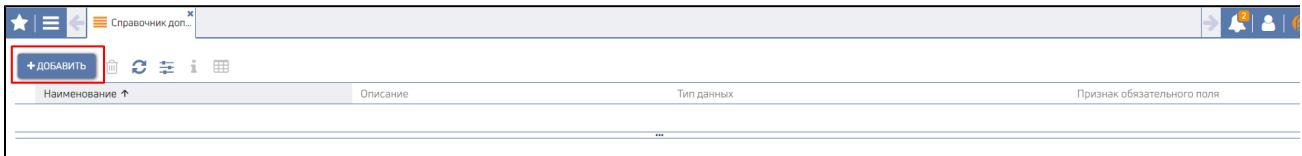
Справочник дополнительных данных пользователей

Если при создании пользователей кроме основных полей (логин, пароль, ФИО, e-mail) необходимо заполнять дополнительную информацию, то следует создать необходимые поля в Справочнике дополнительных данных пользователей.

Добавление данных в список

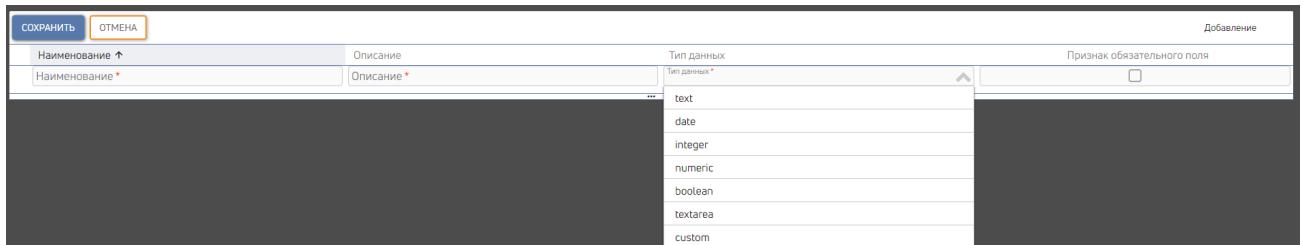
1. перейти на страницу «Справочник дополнительных данных пользователя» (Меню -> CORE -> Авторизация -> Справочник дополнительных данных пользователя);

2. нажать кнопку «Добавить»;



3. заполнить поля;

4. нажать кнопку "Сохранить".



Доступные поля для заполнения:

Наименование поля	Тип данных	Комментарии
Наименование	Текст	Обязательно для заполнения. Будет использоваться в качестве наименования параметра при создании пользователя и в гриде с его данными.
Описание	Текст	Обязательно для заполнения. Будет использоваться в качестве Отображаемого имени поля в окне добавления пользователя и в гриде с его данными.
Тип данных	Выбор из списка	Обязательно для заполнения. Будет использоваться в качестве типа поля в окне добавления пользователя и в гриде с его данными. Доступные значения: <ul style="list-style-type: none">■ boolean;■ custom;■ date;■ integer;■ numeric;■ text;■ textarea
Признак обязательного поля	Да/Нет	

Обратите внимание!

При попытке создать записи с одинаковым значением поля "Наименование" Система вернет ошибку и не выполнит сохранение.

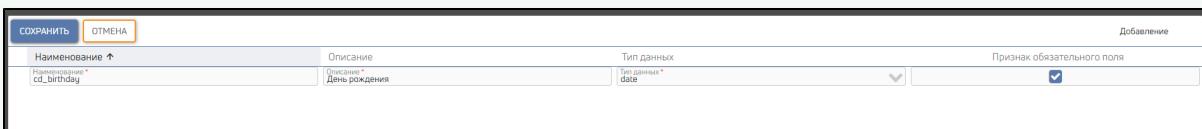
Все созданные в Справочнике данные будут отображаться в окне создания нового пользователя и в Гриде "Данные пользователя" на странице "Пользователи".

Класс поля в окне добавления нового пользователя и редактирования в гриде "Данные пользователя" зависит от выбранного значения поля "Тип данных":

Тип данных	Класс поля	Комментарий
boolean	Field Checkbox	
custom		Используется при необходимости добавления поля сложного класса (например, Combobox, Field Grid и т.д.)
date	Field Date	
integer	Field Integer	
numeric	Field Numeric	
text	Field Text	
textarea	Field Textarea	

Пример

- Добавим в качестве дополнительной информации пользователя дату рождения.



СОХРАНИТЬ ОТМЕНА

Наименование ↑ Описание Тип данных Признак обязательного поля

Наименование * cd_birthday Описание * День рождения Тип данных * date Проверка валидности

- Теперь создадим нового пользователя.



Добавление Пользователя

Логин *

Пароль *

Фамилия *

Имя *

Отчество

E-mail

День рождения *

СОХРАНИТЬ ОТМЕНА

3. Созданное поле "День рождения" с типом "Дата" отображается в окне добавления пользователя и является обязательным для заполнения.

При создания нового пользователя в модификацию данное поле будет передано с параметром "cd_birthday":

```
"data": {
    "ck_id": "",
    "cv_login": "",
    "cv_hash_password": "",
    "cv_surname": "",
    "cv_name": "",
    "cv_patronymic": "",
    "cd_birthday": "2019-10-23T00:00:00"
},
"service": {
    "ck_main": null,
    "ck_page": "D9ED58074C5F4972A5CEEED5B35FFCC7",
    "ck_page_object": "687CEFC7180141B2A0628B8157933438",
    "cl_warning": 0,
    "cv_action": "I"
}
```

4. Данные о дате рождения отобразятся в гриде "Данные пользователя". Для ранее созданных пользователей указанное поле будет пустым:

The screenshot shows a user management interface. At the top, there's a toolbar with buttons for adding, deleting, saving, and filtering. Below it is a grid table with columns: Login, Фамилия (Surname), Имя (Name), Отчество (Middle Name), and E-mail. Three users are listed: admin_core, view_core, and Админ. Below the grid, there are two tabs: 'Роли пользователя' (User Roles) and 'Данные пользователя' (User Data). The 'Данные пользователя' tab is active, showing two fields: 'Описание' (Description) and 'День рождения' (Birthdate). The 'День рождения' field is empty for all users.

5. Заполнить дату рождения для ранее созданных пользователей можно, используя редактирование записи:

This screenshot shows the same user management interface as the previous one, but with a modal window open over the grid. The modal is titled 'Редактирование Данных пользователя' (Edit User Data). It contains a single input field labeled 'День рождения*' (Birthdate*) with a date picker icon. Below the input are two buttons: 'СОХРАНИТЬ' (Save) and 'ОТМЕНА' (Cancel). The background grid shows the same three users. A red box highlights the 'День рождения' input field in the modal.

Данные с типом "custom"

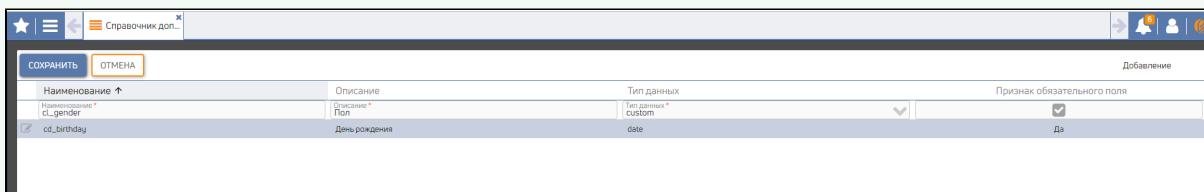
При создании в Справочнике записей с типом "custom" в гриде "Данные пользователя" отобразится соответствующая запись, однако поля в окне редактирования и в окне добавления нового пользователя автоматически не отобразятся.

Для корректного отображения необходимого поля необходимо создать объект нужного класса в соответствующих окнах.

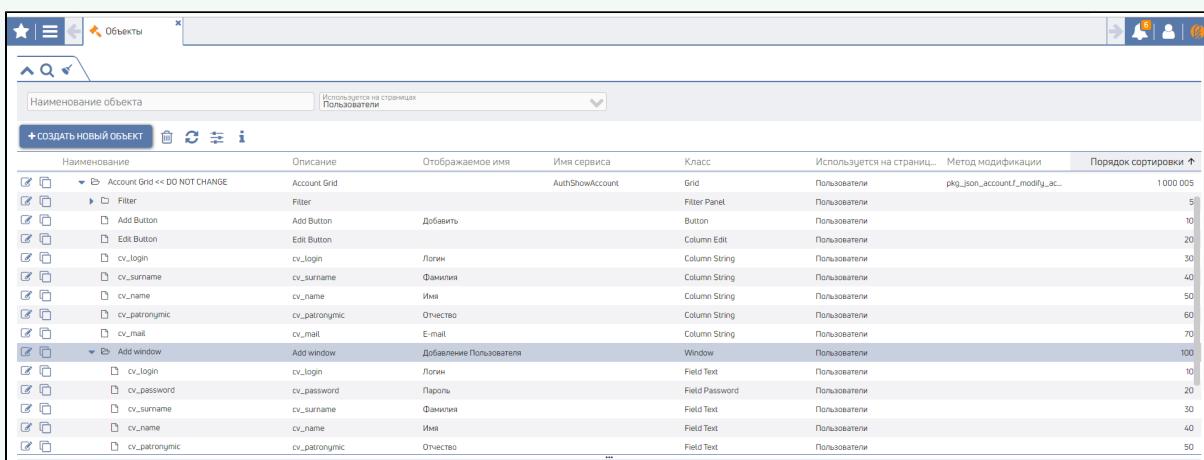
Пример

Необходимо добавить данные о поле пользователя. Для этого следует использовать поле класса Field Combobox.

1. Создаем запись с типом "custom" в Справочнике дополнительных данных пользователя;



2. добавим поле "Выпадающий список" в окно создания нового пользователя.
3. переходим на страницу "Объекты" (Меню CORE Конструктор GUI Объекты), находим окно создание нового пользователя:



4. Создадим поле класса Field Combobox со значением атрибута column = cl_gender.
Также следует указать корректный сервис для данного Field Combobox и значения атрибутов idproperty, valuefield и displayfield.

Наименование	Описание	Тип	Параметры
cv_patronymic	Отчество	Column String	Пользователи
cv_mail	E-mail	Column String	Пользователи
Add window	Добавление Пользователя	Window	Пользователи
cv_login	Логин	Field Text	Пользователи
cv_password	Пароль	Field Password	Пользователи
cv_surname	Фамилия	Field Text	Пользователи
cv_name	Имя	Field Text	Пользователи
cv_patronymic	Отчество	Field Text	Пользователи
cv_mail	E-mail	Field Text	Пользователи
Save Button	Сохранить	Button	Пользователи
AccountInfoField	AuthGetAccountInfoField	Panel Dynamic	Пользователи
Cancel Button	Отмена	Button	Пользователи
Пол	Пол	Field Combobox	
Пользователя		Window	Пользователи

Атрибуты объекта

Наименование	Значение по умолчанию	Значение (переопределено)	Описание
allownew	false		Признак доступности ввода
autoload	true		Признак автозагрузки страницы
column	cl_gender		Наименование колонки в форме
defaultvalue			Значение по умолчанию

5. Обновляем привязку объектов для страницы Пользователи (Меню CORE Конструктор GUI Страницы) и сбрасываем кеш:

Наименование страницы	Код действия просмотра	Код действий
+СОЗДАТЬ СТРАНИЦУ		
Наименование страницы		
Корневой каталог		
CORE		
Конструктор GUI		
Администрирование		
Авторизация		
Действия	515	
Роли	515	
Пользователи	515	
Обновить привязку объектов	516	
Автотесты		
Меню		

Наименование объекта	Наименование класса	Описание	Порядок
Корневой каталог			
Account Grid << DO NOT CHANGE	Grid	Account Grid	10
Account Tabs << DO NOT CHANGE	Tab Panel	Account Tabs	20

6. Переходим на страницу Пользователи и нажимаем кнопку "Добавить".
Поле "Пол" класса Field Combobox успешно добавлено.

Имя	Отчество
Добавление Пользователя	
Логин *	
Пароль *	
Фамилия *	
Имя *	
Отчество	
E-mail	
День рождения *	
Пол	
<input type="button" value="СОХРАНИТЬ"/> <input type="button" value="ОТМЕНА"/>	

7. Аналогично добавляется поле в окно редактирования записей в гриде "Данные пользователя"

Корневой каталог		Название	Описание	Метод	Тип	Компонент
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Account Tabs << DD ..	Account Tabs			Пользователи
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Account Roles Grid	Роли пользователя	AuthShowSelectedRole	Grid	Пользователи
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Account Info Grid	Данные пользователя	AuthShowAccountInfo	Grid	Пользователи
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Edit btn	Edit btn		Column Edit	Пользователи
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	cv_description	Описание		Column String	Пользователи
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	cv_value	Значение		Column String	Пользователи
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Edit window	Данных пользователя		Window	Пользователи
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Save Button	Save Button	Сохранить	Button	Пользователи
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	AccountInfoF...	AccountInfoField	AuthGetAccountInfoFieldEdit	Panel Dynamic	Пользователи
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Cancel Button	Отмена		Button	Пользователи

Сброс провайдера авторизации

Все данные по авторизации (список пользователей, ролей, действий и т.д.) инициализируются платформой при запуске шлюза. Данная информация хранится провайдером авторизация, который перезапускается автоматически каждую ночь.

В случае если:

- была произведена модификация данных авторизации (создание/изменение пользователей, действий и т.д.);
- необходимо, что выполненные изменения применились незамедлительно

необходимо перезапустить провайдер авторизации.

Для этого следует:

1. перейти на страницу «Шлюз» (Меню -> CORE -> Администрирование -> Шлюз);
2. выбрать сервер, для которого будет производиться сброс провайдера авторизации;
3. перейти на вкладку "Настройка провайдеров";
4. в гриде "Провайдеры шлюза" выбрать провайдера авторизации authcore;
5. нажать кнопку "Сброс провайдера" и подтвердить выполнение;

Список провайдеров CORE (база)

Идентификатор	Наименование
admingate	Шлюз
auth	Авторизация
authcore	Авторизация CORE
meta	Метамодель

Провайдеры шлюза

Наименование	Признак загрузки при инициализации	Наименование плагина	Описание	Наст.
admingate	Нет	admingate	Провайдер администрирования	
auth	Да	authmock	Провайдер авторизации	admin
authcore	Да	coreauthpgsql	Провайдер внутренней авторизации	conn
meta	Да	postgresql_db	Провайдер работы с метаданными	core=

Список пользователей, их данные и доступные им действия можно посмотреть во вкладке "Авторизация" на странице "Шлюз" (Меню -> CORE -> Администрирование -> Шлюз):

Список серверов

Наименование сервера	IP адрес сервера
gate	127.0.0.1

Пользователи

Наименование про...	Идентификатор	Логин	Фамилия	Имя	Отчество	Email	Доступные действия
authcore	3dfe75a5-7c82-4d84-9b57-18...	админ	админ	админ			511, 533, 500, 515, 704, 534, 51...
authcore	4fd05ca9-3a9e-4d66-82df-8...	admin_core	admin_core	admin_core			504, 499, 533, 498, 692, 516, 7...
authcore	f167f04b-0a85-4e6f-94df-02a...	view_core	view_core	view_core			497, 511, 499, 704, 491, 503, 69...

Локализация

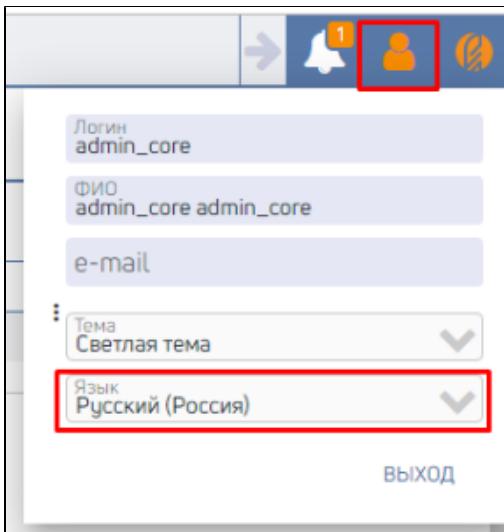
Предусматривает механизм для перевода элементов интерфейса Конструктора на другой язык.

Для настройки Локализации необходимо:

1. добавить нужный язык в Список языков;
2. добавить переводы используемых слов/фраз в Словарь.

Для переключение языка необходимо:

1. нажать на меню Профиля (иконка человечка с правом верхнем углу);
2. выбрать нужный язык из Combobox "Язык";



3. обновить страницу в браузере (нажать кнопку F5).

Словарь

Идентификатор	Слово/Фраза	Тип расположения	Расположение
3e42980262484426035e988...	Атрибут не принадлежит классу...	message	
7f85239199e08c887864d739...	Модуль 3D уже добавлен.	message	
5263808d04a2853701095...	Не заполнено поле 'Признак стат...	message	
c3070343-2949-4908-8707-005...	Системная ошибка: объект не на...	message	
37e6000749238ba706a746...	Не заполнено поле 'Тип атрибути...	message	
7fa6667c3429705016fe0...	Сформирован отчет ID:...	message	
9593787614422196a787ca...	Существует объект '107', сканен...	message	
32102a7c704a44a2120270f...	Небходимо указать признак дс...	message	
39427e95e65e68297c7098...	Без ошибок.	message	
31670668481c3012619418...	Небходимо указать наименован...	message	
387400700800603964030...	Системная ошибка при сохранен...	message	
48154504041d1056a2a5e51...	Не заполнено поле 'Порядок сор...	message	

- Добавление новых значений в список переводимых слов/фраз
- Редактирование слов/фраз
- Удаление слов/фраз
- Добавление перевода слова/фразы
- Редактирование перевода
- Удаление перевода

Добавление новых значений в список переводимых слов/фраз

1. перейти на страницу "Словарь" (CORE Локализация Словарь);
2. все действия производятся в гриде "Список слов/фраз";
3. нажать кнопку "Добавить";

Идентификатор	Слово/Фраза	Тип расположения	Расположение
14acd7f5abb04f00b353906a959f...	45	meta	
3c205218305a4a25bada3700477...	:inputValue не является правильн...	meta	
1583ea7e4b054c75981877121930...	:inputValue не является правильн...	meta	

4. заполнить доступные поля;
Доступные поля для заполнения:

Наименование поля	Тип данных	Комментарии
Слово/Фраза	Текст	Обязательно для заполнения.

Тип расположения	Выбор из списка	Обязательно для заполнения. meta - используется в Конструкторе (например, Отображаемые имена, сообщения при валидации и т.д.); message - используется в тексте ошибок из t_message static - используется в коде
------------------	-----------------	--

5. нажать кнопку "Сохранить".

Идентификатор	Слово/Фраза ↑	Тип расположения	Расположение
	Слово/Фраза *	Тип расположения *	

Обратите внимание!

При создании/редактировании объектов/страниц, значение "Отображаемого наименования" которых отсутствует в Списке слов/фраз, данные значения будут добавлены с Список слов/фраз автоматически при сохранении объекта /страницы.

Обратите внимание!

При добавлении новой записи в грид "Список слов/фраз" автоматически создается запись в подчиненном гриде "Перевод" для языка, выбранного по умолчанию. Подробнее [тут](#)

Редактирование слов/фраз

1. перейти на страницу "Словарь" (CORE Локализация Словарь);
2. все действия производятся в гриде "Список слов/фраз";
3. выбрать слово/фразу для редактирования;

Идентификатор	Слово/Фраза	Тип расположения ↑	Расположение
3ef425b2db2f46b4a2dd243aff188...	Атрибут не принадлежит классу, ...	message	
776f32915eae408ca861964d7d59...	Модуль {0} уже добавлен.	message	

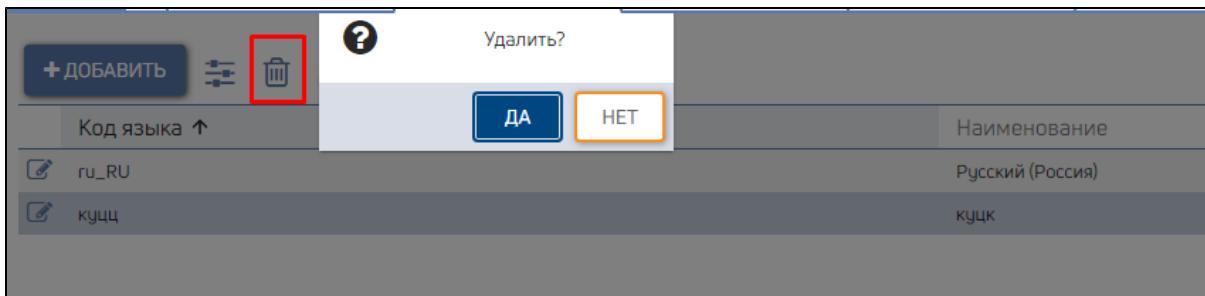
4. нажать кнопку "Редактировать значение" и внести необходимые изменения;
Доступные поля для заполнения:

Наименование поля	Тип данных	Комментарии
Слово/Фраза	Текст	Обязательно для заполнения.
Тип расположения	Выбор из списка	

5. нажать кнопку "Сохранить".

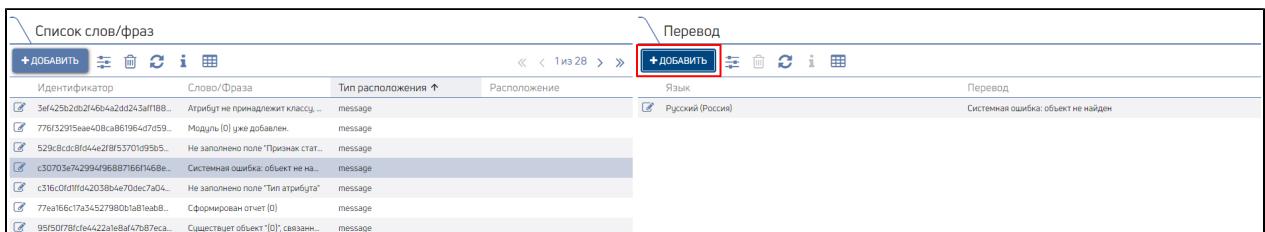
Удаление слов/фраз

1. перейти на страницу "Словарь" (CORE Локализация Словарь);
2. все действия производятся в гриде "Список слов/фраз";
3. выбрать слово/фразу для удаления;
4. нажать кнопку "Удалить" и подтвердить удаление.

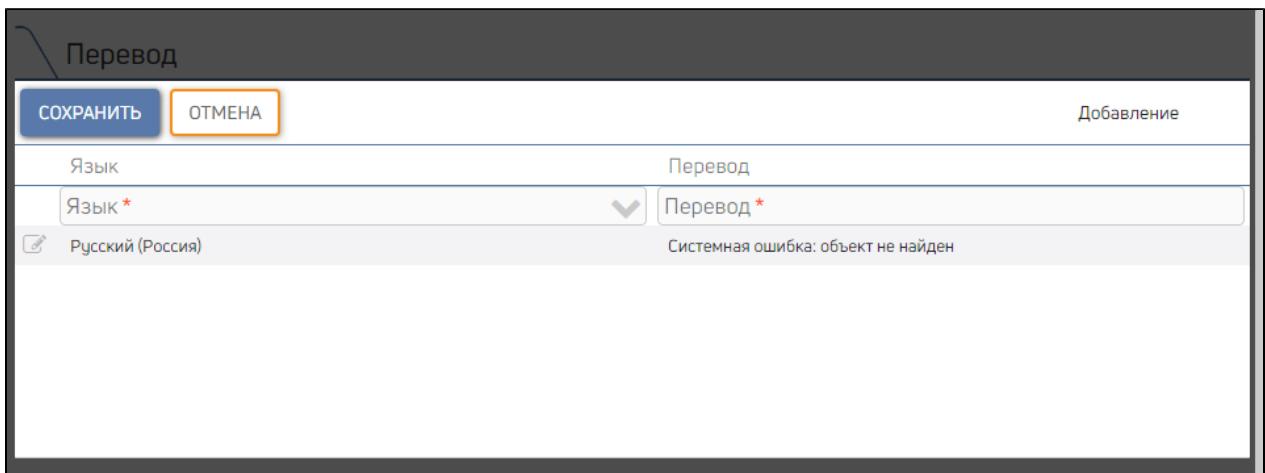


Добавление перевода слова/фразы

1. перейти на страницу "Словарь" (CORE Локализация Словарь);
2. выбрать слово/фразу из грида "Список слов/фраз";
3. дальнейшие действия производятся в гриде "Перевод";
4. нажать кнопку "Добавить";



5. заполнить доступные поля;



Доступные поля для заполнения:

Наименование поля	Тип данных	Комментарии
-------------------	------------	-------------

Язык	Выбор из списка	Обязательно для заполнения.
Перевод	Текст	Обязательно для заполнения.

6. нажать кнопку "Сохранить".

Обратите внимание!

Если для слова/фразы отсутствует перевод на язык, который выбран Пользователем в меню Профиль, то в интерфейсе будет отображен перевод на язык по умолчанию.

Редактирование перевода

1. перейти на страницу "Словарь" (CORE Локализация Словарь);
2. выбрать слово/фразу и перевод для редактирования;
3. все действия производятся в гриде "Перевод";

Словарь				Перевод	
+ добавить	Слово/Фраза	Тип расположения	Расположение	+ добавить	Язык
<input checked="" type="checkbox"/>	Зе42502b02bf4d8442d6245af1f88...	Атрибут не принадлежит классу ...	message	<input checked="" type="checkbox"/>	Русский (Россия)
<input checked="" type="checkbox"/>	776f32915eae408ca861964d7e59...	Модуль (0) уже добавлен.	message		Системная ошибка: объект не найден
<input checked="" type="checkbox"/>	529fc8cd8f644e2f653701995b...	Не заполнено поле 'Примэж стат...	message		
<input checked="" type="checkbox"/>	c30703674299496887667f468e...	Системная ошибка: объект не на...	message		
<input checked="" type="checkbox"/>	c316c0ff1ff4203584e70de7a04...	Не заполнено поле 'Тип атрибуза'	message		
<input checked="" type="checkbox"/>	77ea16617a3452780801a1eef8...	Сформирован отчет [0]	message		

4. нажать кнопку "Редактировать значение" и внести необходимые изменения;

СОХРАНИТЬ
ОТМЕНА

Редактирование

Язык
Перевод

Русский (Россия)

Перевод *
Системная ошибка: объект не найден

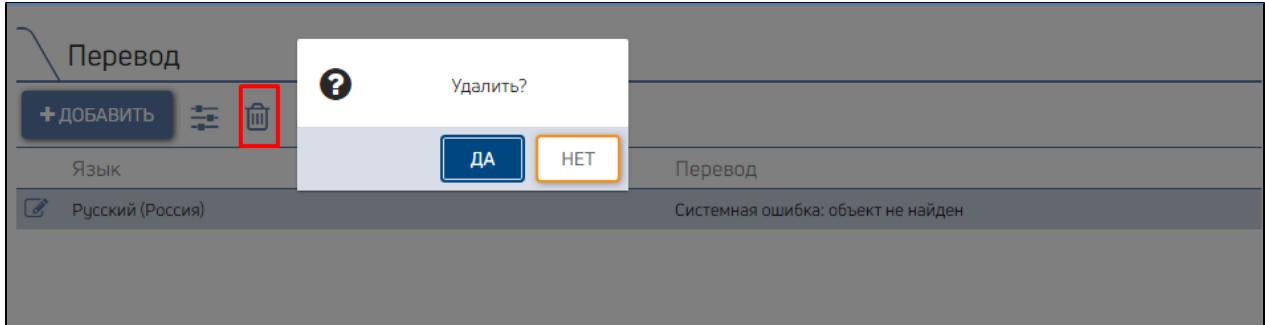
Доступные поля для заполнения:

Наименование поля	Тип данных	Комментарии
Перевод	Текст	Обязательно для заполнения.

5. нажать кнопку "Сохранить".

Удаление перевода

1. перейти на страницу "Словарь" (CORE Локализация Словарь);
2. выбрать слово/фразу и перевод для редактирования;
3. все действия производятся в гриде "Перевод";
4. выбрать перевод для удаления;
5. нажать кнопку "Удалить" и подтвердить удаление.



Список языков

Добавление нового языка в Список языков

- перейти на страницу "Список языков" (CORE Локализация Список языков);
- нажать кнопку "Добавить";

- заполнить доступные поля;

Доступные поля для заполнения:

Наименование поля	Тип данных	Комментарии
Код языка	Текст	Обязательно для заполнения.
Наименование	Текст	Обязательно для заполнения.
По умолчанию	Чек-бокс	

- нажать кнопку "Сохранить".

Обратите внимание!

Языком по умолчанию может быть выбран только один язык.

При создании языка по умолчанию ко всем словам/фразам из Списка слов/фраз будут автоматически сгенерированы переводы на указанный язык со значением: ##Need to translate##

Редактирование языка

1. перейти на страницу "Список языков" (CORE Локализация Список языков);
2. выбрать язык для редактирования;

The screenshot shows a table with two rows. The first row contains 'eng_US' and 'Английский (США)'. The second row contains 'ru_RU' and 'Русский (Россия)'. A red box highlights the 'Редактировать значение' (Edit value) button at the bottom of the table.

Код языка ↑	Наименование
eng_US	Английский (США)
ru_RU	Русский (Россия)

3. нажать кнопку "Редактировать значение" и внести необходимые изменения;
Доступные поля для заполнения:

Наименование поля	Тип данных	Комментарии
Наименование	Текст	Обязательно для заполнения.
По умолчанию	Чек-бокс	

4. нажать кнопку "Сохранить".

Удаление языка

1. перейти на страницу "Список языков" (CORE Локализация Список языков);
2. выбрать язык для удаления;
3. нажать кнопку "Удалить" и подтвердить удаление.

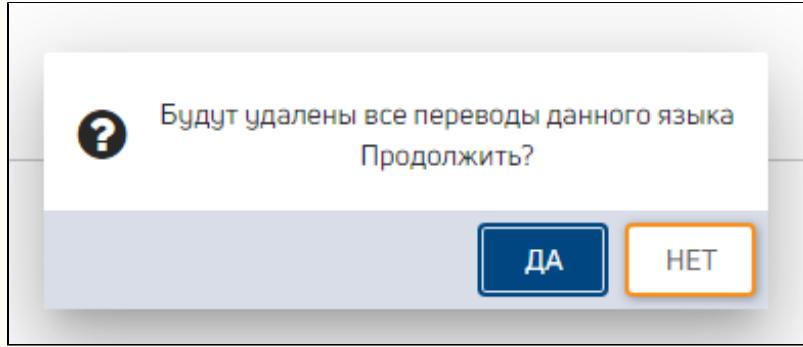
The screenshot shows a table with two rows. The first row contains 'ru_RU' and 'Русский (Россия)'. The second row contains 'куцк' and 'куцк'. A red box highlights the delete icon in the toolbar. A confirmation dialog box titled 'Удалить?' (Delete?) is overlaid on the screen, with 'ДА' (Yes) and 'НЕТ' (No) buttons.

Код языка ↑	Наименование
ru_RU	Русский (Россия)
куцк	куцк

Обратите внимание!

При удалении языка из Словаря будут автоматически удалены все переводы на данный язык.

Система отобразит сообщение-подтверждение:



Патчинг

Модуль патчинга позволяет выполнять перенос данных с одного сервера CORE на другой. например, переносить новые страницы или пользователей с тестового сервера на боевой.

Платформа CORE позволяет выполнять патчинг:

- метаданных (страницы/каталоги/модули, ошибки, сервисы, локализация);
- внутренняя авторизация (действия, роли, пользователи);
- интеграционные сервисы.

Патчинг авторизации

Данный тип сборки доступен только после развертывания
БД внутренней авторизации. [Подробнее](#)

Позволяет создавать патчи со следующими данными из БД
внутренней авторизации:

- Создание плагина
- Создание сборки
- Состав сборки
- Установка

- Действия;
- Роли;
- Справочник дополнительных данных пользователя;
- Пользователи.

Создание плагина

Для создания сборок необходимо создать плагин патчера.

1. Перейти на страницу "Шлюз" (CORE Администрирование Шлюз);
2. Выбрать сервер, для которого будет создан плагин;
3. Перейти на вкладку "Плагины" и нажать кнопку "Добавить";
4. Заполнить поля в появившемся окне:

Добавление плагина

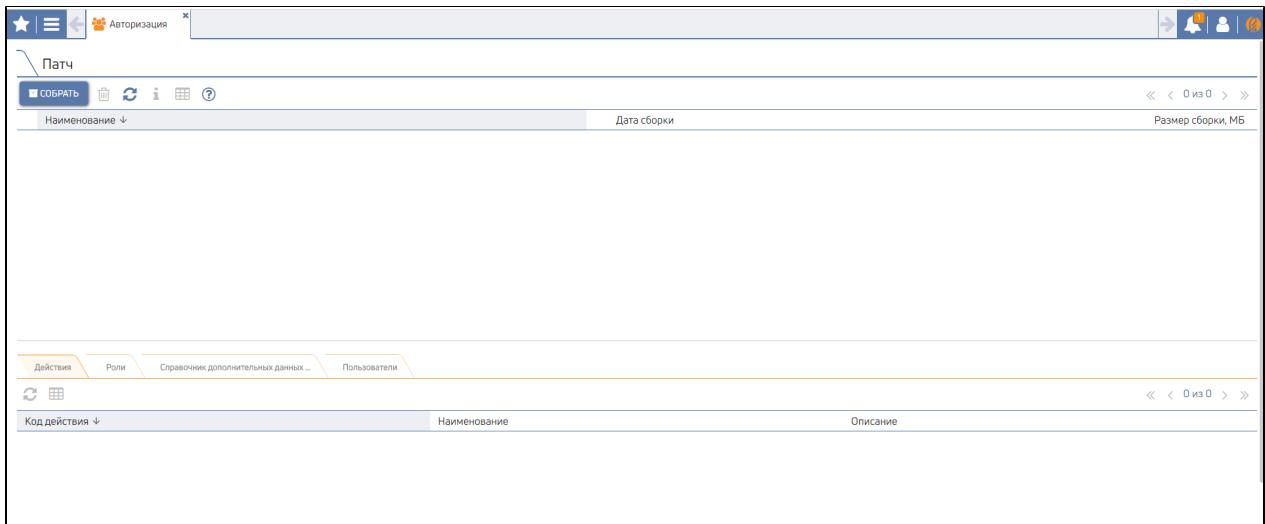
Наименование *	create_patch
Провайдер *	authcore
Наименование плагина *	patcher
Применять всегда	<input type="checkbox"/>
Порядок *	1
Применять если не указаны другие плагины	<input type="checkbox"/>
Описание	Создание патча для авторизации
dirlaws riak dir	
Адрес Riak Dir Aws	/opt/patch
Наименование корзины S3	
Id key S3 Storage	
Secret key S3 Storage	
СОХРАНИТЬ ОТМЕНА	

5. Нажать кнопку "Сохранить";
6. Перезапустить шлюз.

Создание сборки

Для создания патча необходимо:

1. Перейти на страницу "Авторизация" (CORE Патчинг Авторизация);



2. Нажать кнопку "Собрать";
3. в появившемся окне выбрать состав патча и нажать кнопку "Сохранить";

Состав сборки

Действия	0 / 1*	<input type="button" value=""/>
Роли		<input type="button" value=""/>
Справочник доп. данных пользователя		<input type="button" value=""/>
Пользователи		<input type="button" value=""/>

4. Будет сформирован архив сборки с наименованием вида "Patch_дата_время.zip" (например: Patch_2020-02-07_14:15:26.zip)
5. Скачать архив можно, нажав на кнопку:



6. Состав сборки отображается в дочерних гридах:

The screenshot shows a software interface titled 'Патч' (Patch). At the top, there are buttons for 'СОБРАТЬ' (Collect) and other file operations. Below this is a header row with columns for 'Наименование' (Name) and 'Дата сборки' (Build Date). A single entry is listed: 'Patch_2020-02-07_14:15:26.zip' with the date '07.02.2020 00:00:00'. Below the header is a navigation bar with tabs: 'Действия' (Actions), 'Роли' (Roles), 'Справочник дополнительных данных...' (Additional Data Dictionary), and 'Пользователи' (Users). The 'Действия' tab is highlighted with a red border. The main area contains a table with columns 'Код действия' (Action Code), 'Наименование' (Name), and 'Описание' (Description). The table entries are:

Код действия	Наименование	Описание
492	Доступ на модификацию страницы "Атрибуты"	Доступ на модификацию страницы "Атрибуты"
499	Доступ на просмотр страницы "Объекты"	Доступ на просмотр страницы "Объекты"
503	Доступ на просмотр страницы "Профиль"	Доступ на просмотр страницы "Профиль"
504	Доступ на модификацию страницы "Профиль"	Доступ на модификацию страницы "Профиль"

Состав сборки

В зависимости от выбора параметров сборки в архиве Patch*.zip папке /auth будут сформированы файлы:

Что выбрано	Наименование файла	Что будет выгружено
Действия	Action.sql Action.sql.md5	<ul style="list-style-type: none">сами действиясвязь роли и действия
Роли	Role.sql Role.sql.md5	<ul style="list-style-type: none">сами ролисвязь роли и действиясвязь роли и пользователя
Справочник дополнительных данных пользователя	Info.sql Info.sql.md5	<ul style="list-style-type: none">сам справочниксвязь доп.данных и пользователя
Пользователи	Account.sql Account.sql.md5	<ul style="list-style-type: none">сами пользователироли и пользователя

Установка

Для установки сборки необходимо:

- распаковать архив сборки Patch_*.zip;
- в файле liquibase.properties указать корректные адрес и порт БД;

```
driver: org.postgresql.Driver
url: jdbc:postgresql://host:port/core_auth
username: s_su
password: s_su
```

- запустить файл update.

Патчинг метаданных

Патчинг метаданных позволяет экспорттировать следующие данные:

- модули/каталоги/страницы;
- сервисы;
- ошибки;
- словари локализации.

- Создание плагина
- Создание сборки
- Состав сборки
- Установка

Создание плагина

Для создание сборок необходимо создать плагин патчера.

1. Перейти на страницу "Шлюз" (CORE Администрирование Шлюз);
2. Выбрать сервер, для которого будет создан плагин;
3. Перейти на вкладку "Плагины" и нажать кнопку "Добавить";
4. Заполнить поля в появившемся окне:

Добавление плагина

Наименование*	create_patch
Провайдер*	meta
Наименование плагина*	patcher
Применять всегда	<input type="checkbox"/>
Порядок*	1
Применять если не указаны другие плагины	<input type="checkbox"/>
Описание	dirlaws riak dir
Адрес Riak Dir Aws	/opt/patch
Наименование корзины S3	
Id key S3 Storage	
Secret key S3 Storage	
<input type="button" value="СОХРАНИТЬ"/> <input type="button" value="ОТМЕНА"/>	

5. Нажать кнопку "Сохранить";

6. Перезапустить шлюз.

Список серверов

+ Добавить

Наименование сервера ↑

Создание сборки

Для создания патча необходимо:

1. Перейти на страницу "Мета" (CORE Патчинг Мета);

Мета

Патч

Собрать

Наименование

Дата сборки

Размер сборки, МБ

Модули/Каталоги/Страницы Сервисы Ошибки Язык локализации

2. Нажать кнопку "Собрать";
3. в появившемся окне выбрать состав патча и нажать кнопку "Сохранить";

Состав сборки

0 / 1*

Модуль/Каталоги/Страницы

Сервисы

Ошибки

Язык локализации

СОХРАНИТЬ

ОТМЕНА

4. Будет сформирован архив сборки с наименованием вида "Patch_дата_время.zip" (например: Patch_2020-02-07_11:24:37.zip)
5. Скачать архив можно, нажав на кнопку:

Состав сборки отображается в дочерних гридах:

Наименование: Patch_2020-02-07_11:24:37.zip
Дата сборки: 07.02.2020

Модули/Каталоги/Страницы Сервисы Ошибки Язык локализации

Наименование
Объекты
Автотесты
Сценарии
Сервисы
Атрибуты
Страницы
Приложения
УСПО
Авторизация
Ресурсы

Состав сборки

В зависимости от выбора параметров сборки в архиве Patch*.zip папке /meta будут сформированы файлы:

Что выбрано	Наименование файла	Что будет выгружено
Страница /Каталог /Модуль	Page_"Ид_страницы".sql Page_"Ид_страницы".sql.md5	<ul style="list-style-type: none"> Сама страница и ее действия просмотра/редактирования; Каталог/модуль, содержащие выбранную страницу; все объекты, которые имеются на странице, их атрибуты, мастер-связи объектов; глобальные переменные на странице; переводы слов/фраз, используемых на странице, на все языки, которые есть на сервере, с которого снимается патч + сами языки
	Query_"Наименование_провайдера".sql	Сервисы, использующиеся на выгружаемой странице

	Query_" Наименование_провайдера".sql.md5	
Сервис	Query_" Наименование_провайдера".sql Query_" Наименование_провайдера".sql.md5	Выбранные сервисы из t_query
Ошибки	Message.sql Message.sql.md5	<ul style="list-style-type: none"> • Выбранные ошибки из t_message • переводы слов/фраз из ошибок на все языки, которые есть на сервере, с которого снимается патч + сами языки
Язык локализации	Localization_"Код_языка".sql Localization_"Код_языка".sql.md5	<ul style="list-style-type: none"> • выбранный язык • все переводы слов и фраз для выбранного языка

Установка

Для установки сборки необходимо:

1. распаковать архив сборки Patch_*.zip;
2. в файле liquibase.properties указать корректные адрес и порт БД;

```
driver: org.postgresql.Driver
url: jdbc:postgresql://host:port/core
username: s_su
password: s_su
```

3. запустить файл update.