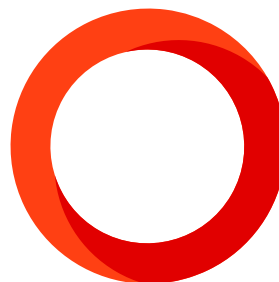
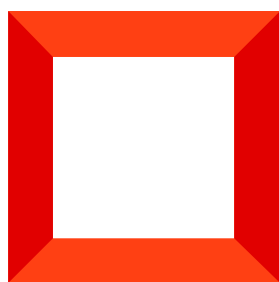
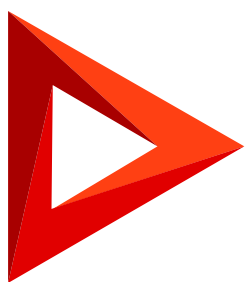


# Creatio IDE

Версия 8.0



Эта документация предоставляется с ограничениями на использование и защищена законами об интеллектуальной собственности. За исключением случаев, прямо разрешенных в вашем лицензионном соглашении или разрешенных законом, вы не можете использовать, копировать, воспроизводить, переводить, транслировать, изменять, лицензировать, передавать, распространять, демонстрировать, выполнять, публиковать или отображать любую часть в любой форме или посредством любые значения. Обратный инжиниринг, дизассемблирование или декомпиляция этой документации, если это не требуется по закону для взаимодействия, запрещены.

Информация, содержащаяся в данном документе, может быть изменена без предварительного уведомления и не может гарантировать отсутствие ошибок. Если вы обнаружите какие-либо ошибки, сообщите нам о них в письменной форме.

# Содержание

<b>Creatio IDE</b>	<b>4</b>
Настроить доступ к разделу Конфигурация	4
Открыть раздел Конфигурация	4
Интерфейс раздела Конфигурация	5
<b>Создать пользовательское действие процесса</b>	<b>9</b>
1. Создать схему действия процесса	9
2. Добавить параметры пользовательского действия	11
3. Добавить логику пользовательского действия	12
4. Выполнить тестирование	13
Результат выполнения примера	20
Добавить действие процесса на вкладку [Элементы процесса]	21
<b>Операции в Creatio IDE</b>	<b>23</b>
Управлять файловой системой	23
Управлять пакетами	24
Управлять хранилищами SVN	24
Управлять конфигурационными элементами	25
Управлять исходным кодом	34
Управлять конфигурацией	35
<b>Клиентский модуль</b>	<b>37</b>
Реализовать невизуальный модуль	38
Реализовать визуальный модуль	39
Реализовать замещающий модуль	44
<b>Объект</b>	<b>46</b>
Реализовать объект	47
Реализовать замещающий объект	54
Деактивировать записи объекта	55
<b>Исходный код (C#)</b>	<b>56</b>
Реализовать исходный код	57
Реализовать замещающий класс	58
<b>Замещение конфигурационных элементов</b>	<b>59</b>

# Creatio IDE



**Creatio IDE** — встроенная среда разработки на платформе Creatio. Creatio IDE реализована в виде раздела [ Конфигурация ] ([ *Configuration* ]). **Назначение** раздела [ Конфигурация ] ([ *Configuration* ]) — выполнение операций со структурными элементами, которые позволяют разрабатывать функциональность.

**Структурные элементы**, операции с которыми позволяет выполнять Creatio IDE:



- Файловая система.
- Пакеты.
- Хранилища SVN.
- Конфигурационные элементы.
- Исходный код.
- Конфигурация.

Подробнее читайте в статье [Операции в Creatio IDE](#).

## Настроить доступ к разделу [ Конфигурация ]

Доступ к разделу [ Конфигурация ] ([ *Configuration* ]) настраивается на уровне системных операций. Если пользователь не имеет доступа к разделу [ Конфигурация ] ([ *Configuration* ]), то отображается стандартное сообщение с указанием операции и недостающих прав. По умолчанию доступ к основным системным операциям имеют только администраторы приложения. Creatio предоставляет возможность настройки доступа к системным операциям для пользователей или групп пользователей. Подробнее читайте в статье [Настроить права доступа на системные операции](#).

Чтобы **настроить доступ к разделу** [ Конфигурация ] ([ *Configuration* ]):


1. Перейдите в дизайнер системы по кнопке . В блоке [ Пользователи и администрирование ] ([ *Users and administration* ]) перейдите по ссылке [ Права доступа на операции ] ([ *Operation permissions* ]).
2. Выберите системную операцию [ Доступ к разделу "Конфигурация" ] ([ *Can manage configuration elements* ]), код `CanManageSolution`.
3. На детали [ Доступ к операции ] ([ *Operation permission* ]) нажмите  и укажите получателя прав.

В результате запись отобразится на детали [ Доступ к операции ] ([ *Operation permission* ]) в колонке [ Уровень доступа ] ([ *Access level* ]) со значением [ Да ] ([ *Yes* ]). Пользователи, которые входят в указанную роль, получают доступ к системной операции [ Доступ к разделу "Конфигурация" ] ([ *Can manage configuration elements* ]), код `CanManageSolution`.

## Открыть раздел [ Конфигурация ]

**Способы перехода** в раздел [ Конфигурация ] ([ *Configuration* ]) для приложения Creatio на платформе

## .NET Framework:

- По кнопке  через дизайнер системы. В блоке [ *Конфигурирование разработчиком* ] ([ *Admin area* ]) перейдите по ссылке [ *Управление конфигурацией* ] ([ *Advanced settings* ]).
- По ссылке [Адрес приложения Creatio]/0/ClientApp/#/WorkspaceExplorer . Например, `http://my.creatio.com/0/ClientApp/#/WorkspaceExplorer` .
- По алиасу `/we` . Например, `http://my.creatio.com/0/we` .
- По алиасу `/conf` . Например, `http://my.creatio.com/0/conf` .
- По алиасу `/dev` . Например, `http://my.creatio.com/0/dev` .

Для приложения Creatio на платформе **.NET Core** способы перехода в раздел аналогичны. Отличие — не нужно использовать приставку `/0` .

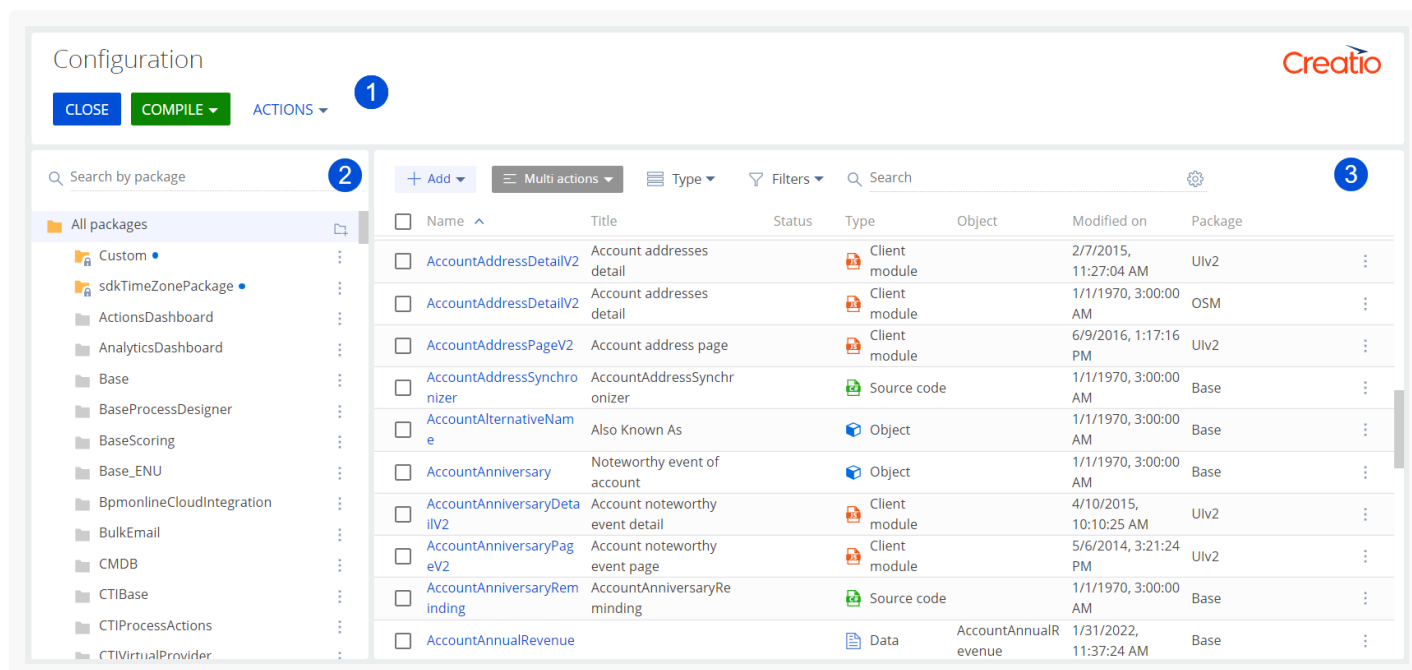
В результате раздел [ *Конфигурация* ] ([ *Configuration* ]) открывается в новой вкладке.

## Интерфейс раздела [ *Конфигурация* ]

**Функциональные области** интерфейса раздела [ *Конфигурация* ] ([ *Configuration* ]):

- Панель инструментов (1).
- Область работы с пакетами (2).
- Рабочая область (3).

Интерфейс раздела [ *Конфигурация* ] ([ *Configuration* ]) представлен на рисунке ниже.



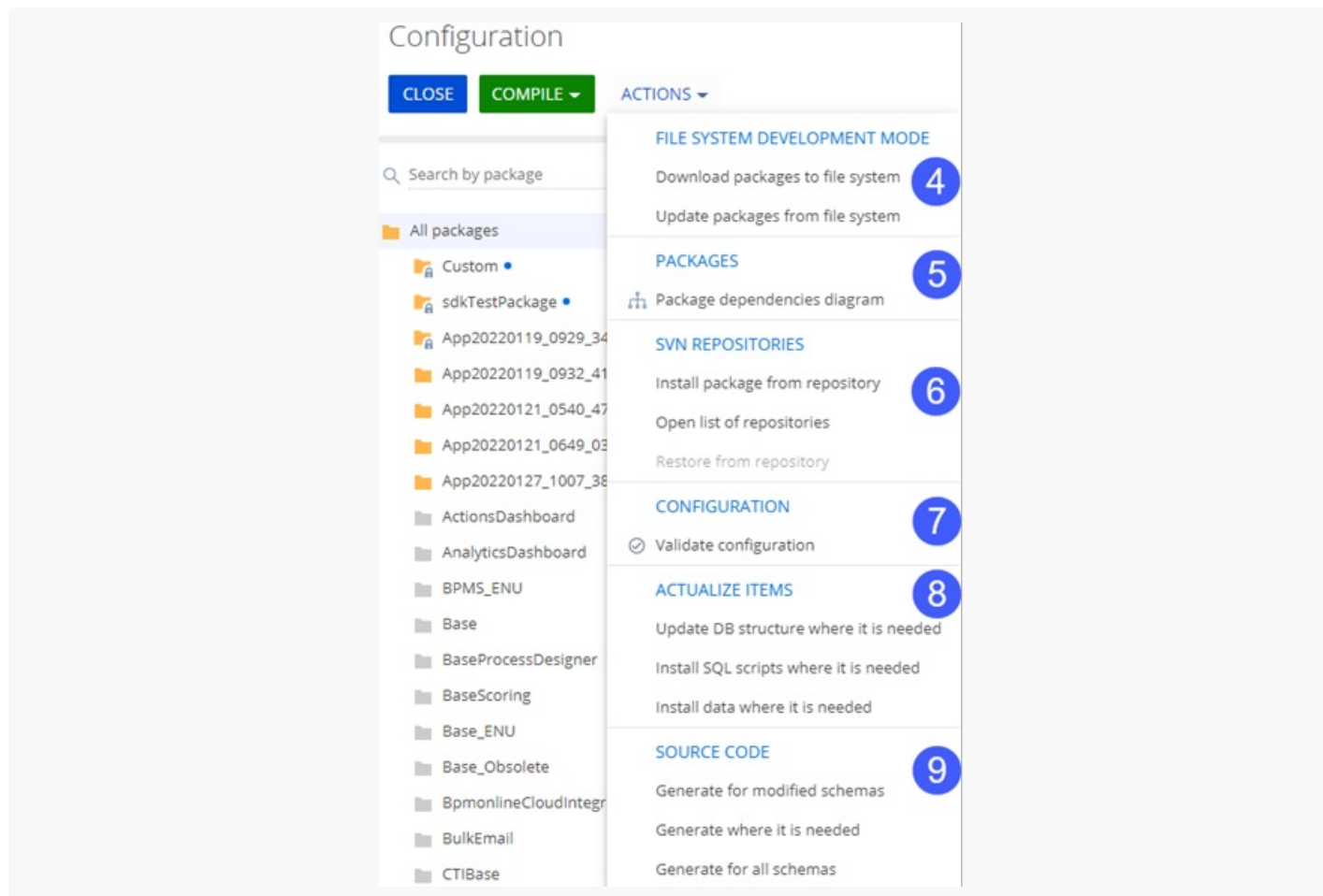
## Панель инструментов (1)

**Панель инструментов (1)** позволяет:

- Закрывать раздел [ Конфигурация ] ([ *Configuration* ]) (кнопка [ Закрыть ] ([ *Close* ])).
- Компилировать всю конфигурацию (кнопка [ Компилировать ] ([ *Compile* ]) и пункт [ Перекомпилировать все ] ([ *Compile all* ]) выпадающего списка кнопки [ Компилировать ] ([ *Compile* ])).
- Управлять условиями разработки (выпадающий список [ Действия ] ([ *Actions* ])).

Управлять условиями разработки позволяют пункты групп выпадающего списка [ Действия ] ([ *Actions* ]) панели инструментов (1).

Группы действий выпадающего списка [ Действия ] ([ *Actions* ]) панели инструментов (1) представлены на рисунке ниже.






**Группы действий**, которые позволяют управлять условиями разработки:

- [ Разработка в файловой системе ] ([ *File system development mode* ]) (4).
- [ Пакеты ] ([ *Packages* ]) (5).
- [ Хранилища SVN ] ([ *SVN repositories* ]) (6).
- [ Проверить конфигурацию ] ([ *Configuration* ]) (7).
- [ Актуализировать элементы ] ([ *Actualize items* ]) (8).
- [ Исходный код ] ([ *Source code* ]) (9).

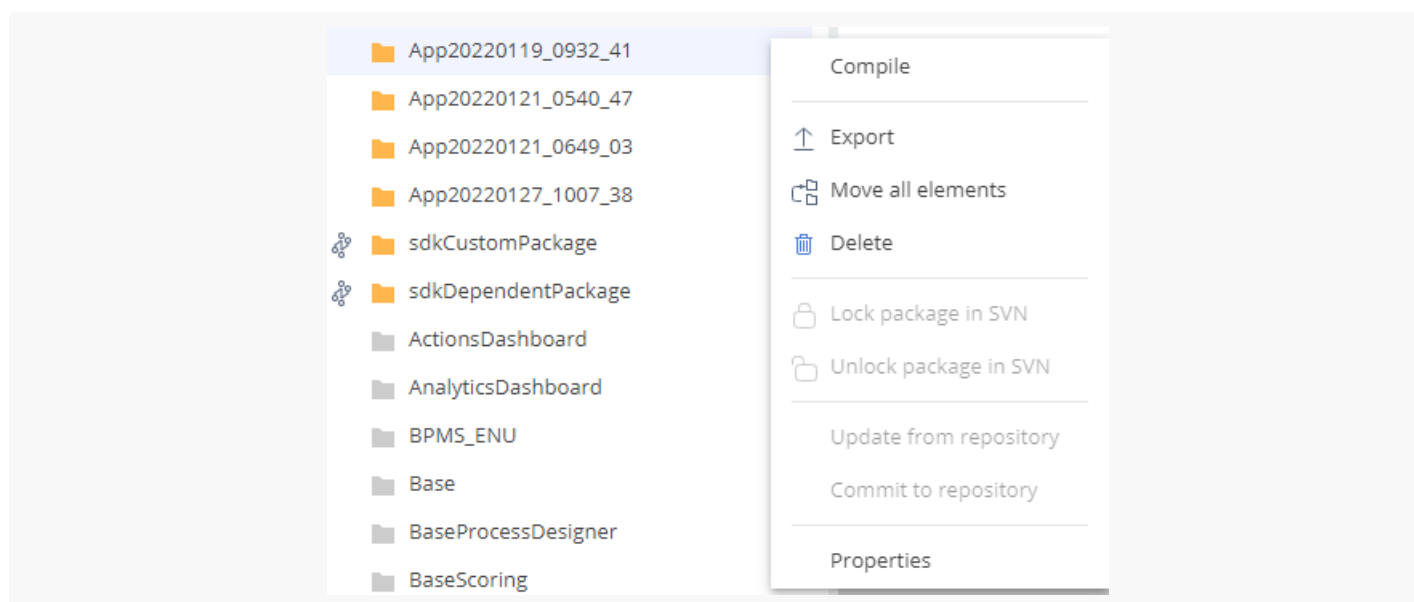
## Область работы с пакетами (2)

**Область работы с пакетами (2)** позволяет:

- Выполнить поиск пакета по его имени (строка поиска [ *Поиск по пакетам* ] ([ *Search by package* ])).
- Создать пакет (кнопка ). При нажатии на кнопку отображается окно создания нового пакета, которое позволяет задать название и описание пакета, настроить компиляцию создаваемого пакета в пакет-сборку, добавить зависимости, а также указать SVN-хранилище. Чтобы **создать пакет**, воспользуйтесь инструкцией, которая приведена в статье [Создать пользовательский пакет](#).
- Посмотреть перечень пакетов приложения (группа [ *Все пакеты* ] ([ *All packages* ])). Пакеты отображаются в алфавитном порядке. Вверху перечня отображаются измененные пакеты и пакеты, которые доступны для редактирования. При выборе текущей группы конфигурационные элементы всех пакетов приложения отображаются в алфавитном порядке в реестре раздела [ *Конфигурация* ] ([ *Configuration* ]). При выборе единичного пакета в алфавитном порядке отображаются конфигурационные элементы текущего пакета. Измененные пакеты отображаются вверху перечня пакетов группы [ *Все пакеты* ] ([ *All packages* ]) и содержат символ  возле имени пакета.

**Меню пакета** области работы с пакетами (2) позволяет выполнять действия с текущим пакетом. Чтобы **открыть меню пакета**, нажмите на  в строке с названием пакета.

Меню пакета представлено на рисунке ниже.



**Меню пакета** позволяет:

- Компилировать пакет в пакет-сборку (пункт [ *Компилировать* ] ([ *Compile* ])). Пункт неактивен для пакетов, в свойствах которых не установлен признак [ *Компилировать в отдельную сборку* ] ([ *Compile into a separate assembly* ]). Этот признак указывает, что пакет является пакетом-сборкой. Подробнее о пакете-сборке читайте в статье [Пакет-сборка](#).
- Выгрузить пакет в \*.zip-архив (пункт [ *Экспортировать* ] ([ *Export* ])). Используется для экспорта пакета с целью дальнейшей установки на другую [рабочую среду](#). Подробнее об экспорте пакета читайте в статье [Перенести пакеты](#).
- Переместить все конфигурационные элементы пакета в другой пакет (пункт [ *Переместить все*

элементы ] ([ *Move all elements* ])). После выбора пункта отображается окно выбора пакета, в который планируется переместить конфигурационные элементы. При перемещении конфигурационных элементов Creatio IDE предоставляет возможность автоматической установки зависимостей.

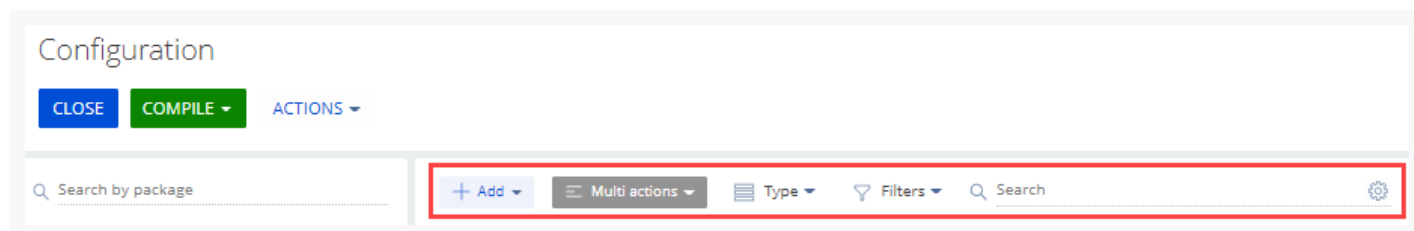
- Удалить пакет (пункт [ *Удалить* ] ([ *Delete* ])). Пункт неактивен для предустановленных пакетов. Creatio IDE позволяет удалить пустые пакеты и пакеты с конфигурационными элементами, которые не являются родительскими. При попытке удалить пакет с родительскими конфигурационными элементами отображается перечень зависимых пакетов и элементов, зависимых от конфигурационных элементов удаляемого пакета, которые препятствуют удалению.
- Заблокировать от изменений пакет в подключенном SVN-хранилище (пункт [ *Заблокировать пакет в SVN* ] ([ *Lock package in SVN* ])). Пункт активен только для пакетов, которые установлены с SVN-хранилища.
- Разблокировать для изменений пакет в подключенном SVN-хранилище (пункт [ *Разблокировать пакет в SVN* ] ([ *Unlock package in SVN* ])). Пункт активен только для предварительно заблокированных пакетов, которые установлены с SVN-хранилища.
- Обновить пакет из подключенного хранилища системы контроля версий (пункт [ *Обновить из хранилища* ] ([ *Update from repository* ])). Обновление пакета описано в статье [Контроль версий в Creatio IDE](#).
- Зафиксировать пакет в подключенном хранилище системы контроля версий (пункт [ *Зафиксировать в хранилище* ] ([ *Commit to repository* ])). Фиксация пакета в системе контроля версий описана в статье [Контроль версий в Creatio IDE](#).
- Посмотреть свойства пакета (пункт [ *Свойства* ] ([ *Properties* ])). Вкладка [ *Свойства пакета* ] ([ *Package properties* ]) позволяет настроить зависимости текущего пакета (если пакет доступен для редактирования). Также вкладка содержит системную информацию: кто создал и отредактировал пакет, даты создания и изменения, уникальный идентификатор, первичный ключ пакета в таблице базы данных. Чтобы **открыть свойства пакета**, дважды кликните по имени пакета.

## Рабочая область (3)

**Инструменты** рабочей области (3):

- Панель инструментов.
- Реестр раздела [ *Конфигурация* ] ([ *Configuration* ]).
- Меню конфигурационного элемента.

**Панель инструментов** рабочей области (3) раздела [ *Конфигурация* ] ([ *Configuration* ]) представлена на рисунке ниже.




**Панель инструментов** рабочей области (3) позволяет:



- Выбрать вид добавляемого конфигурационного элемента (выпадающий список [ *Добавить* ] ([ *Add* ])).
- Загрузить в пользовательский пакет схему (\*.md) или внешнюю сборку (\*.dll) (пункт [ *Импортировать* ] ([ *Import* ]) выпадающего списка [ *Добавить* ] ([ *Add* ])).
- Выполнить массовые действия с конфигурационными элементами (выпадающий список [ *Массовые действия* ] ([ *Multi actions* ])).
- Выбрать тип отображаемых конфигурационных элементов (выпадающий список [ *Тип* ] ([ *Type* ])).
- Выбрать состояние отображаемых конфигурационных элементов (выпадающий список [ *Фильтры* ] ([ *Filters* ])).
- Выполнить поиск конфигурационного элемента по его имени (строка поиска [ *Поиск* ] ([ *Search* ])).

**Реестр раздела** [ *Конфигурация* ] ([ *Configuration* ]) рабочей области (3) позволяет отобразить перечень конфигурационных элементов.

**Меню конфигурационного элемента** рабочей области (3) позволяет выполнить единичное действие с конфигурационным элементом. Чтобы **вызвать меню конфигурационного элемента**, в строке конфигурационного элемента реестра раздела нажмите на . Перечень пунктов меню конфигурационного элемента зависит от типа конфигурационного элемента.

# Создать пользовательское действие процесса



Средний

Во время работы с бизнес-процессами в Creatio возникает необходимость выполнять однотипные операции. Для этих целей используется элемент [ *Выполнить действие процесса* ] ([ *User Task* ]), для которого существует возможность выбрать наиболее подходящий в конкретной ситуации тип действия — [ *Пользовательское действие* ] ([ *User Task* ]). Подробнее об элементе [ *Выполнить действие процесса* ] ([ *User Task* ]) можно узнать из статьи [Элемент процесса \[ \*Выполнить действие процесса\* \]](#).

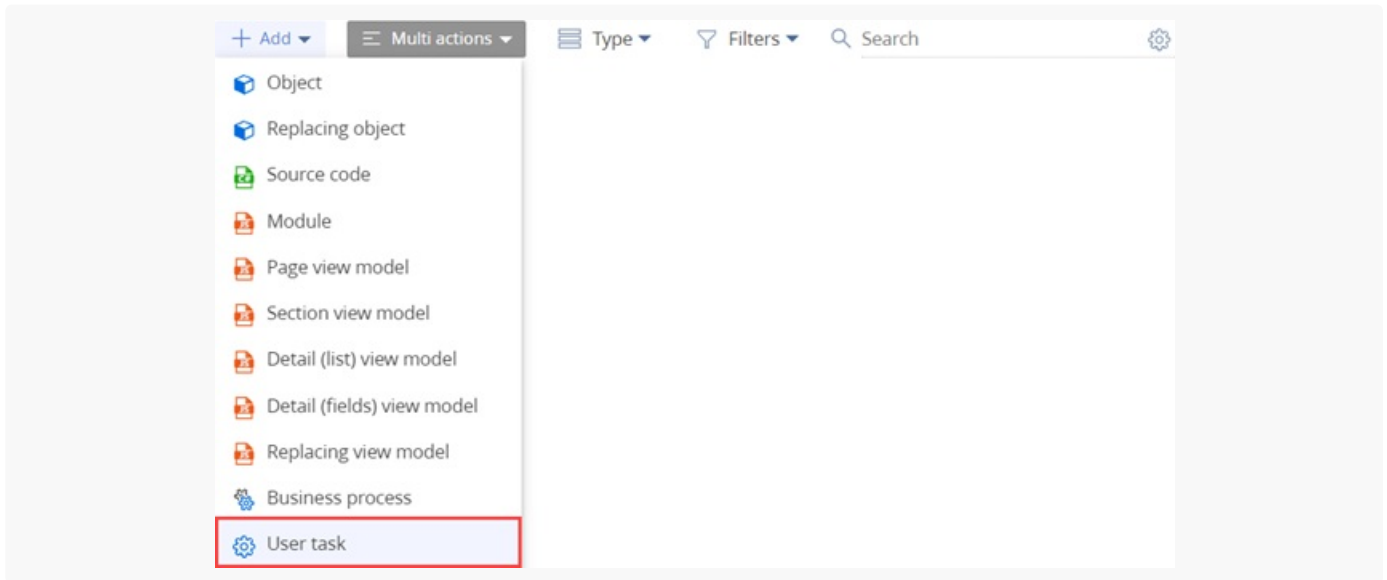
По умолчанию в системе доступен набор преднастроенных пользовательских действий, однако, могут возникнуть ситуации, когда для выполнения определенного бизнес-процесса необходимо создать новое пользовательское действие.

Создать новое пользовательское действие можно с помощью конфигурационной схемы [ *Действие процесса* ] ([ *User Task* ]). В простой реализации действие процесса частично повторяет логику элемента процесса [ *Задание-сценарий* ] ([ *Script Task* ]). При этом, созданное пользовательское действие можно многократно использовать в разных процессах. При внесении изменений в пользовательское действие, эти изменения будут применены ко всем процессам, в которых используется текущее пользовательское действие.

**Пример.** Создать простое пользовательское действие процесса, которое вычисляет сумму двух чисел. Для вычисления суммы используйте два числа, которые задаются в качестве параметров действия.

## 1. Создать схему действия процесса

1. [Перейдите в раздел \[ Конфигурация \]](#) ([ *Configuration* ]) и выберите пользовательский [пакет](#), в который будет добавлена схема.
2. На панели инструментов реестра раздела нажмите [ *Добавить* ] —> [ *Действие процесса* ] ([ *Add* ] —> [ *User Task* ]).



3. В дизайнерах схем заполните свойства схемы:

- [ *Код* ] ([ *Code* ]) — "UsrSampleProcessUserTask".
- [ *Заголовок* ] ([ *Title* ]) — "User Task Sample".

**User task**

Code \*  
UsrSampleProcessUserTask

Title \*  
User Task Sample

Package \*  
sdkCreatingUserTask

Description

Parameters edit page

Dcm parameters edit page

Color  
#839DC3

Small vector image

Large vector image

CANCEL APPLY

Для применения заданных свойств нажмите [ Применить ] ([ Apply ]).

## 2. Добавить параметры пользовательского действия

**Назначение** параметров пользовательского действия — вернуть результат выполнения действия.

Чтобы **добавить параметры**:

1. В узле [ Параметры ] ([ Parameters ]) нажмите кнопку **+**.
2. Заполните **свойства параметра**:
  - [ Код ] ([ Code ]) — "FirstNumber".
  - [ Заголовок ] ([ Title ]) — "First number".
  - [ Тип ] ([ Type ]) — выберите "Целое" ("Integer").
  - Установите признак [ Сериализуемый ] ([ Serializable ]).

3. Добавьте второй параметр пользовательского действия:

- [ Код ] ([ Code ]) — "SecondNumber".
- [ Заголовок ] ([ Title ]) — "Second number".
- [ Тип ] ([ Type ]) — выберите "целое" ("Integer").
- Установите признак [ Сериализуемый ] ([ Serializable ]).

4. Добавьте третий параметр пользовательского действия:

- [ Код ] ([ Code ]) — "SumOfNumbers".
- [ Заголовок ] ([ Title ]) — "Sum of numbers".
- [ Тип ] ([ Type ]) — выберите "целое" ("Integer").
- Установите признак [ Сериализуемый ] ([ Serializable ]).

### 3. Добавить логику пользовательского действия

Логика работы пользовательского действия задается в методе `InternalExecute()` автогенерируемого исходного кода схемы пользовательского действия.

1. В дизайнера схемы измените метод `InternalExecute()` для реализации необходимой бизнес-логики.

**Метод** InternalExecute()

```
protected override bool InternalExecute(ProcessExecutingContext context) {
    // Выполнение операций с параметрами действия.
    SumOfNumbers = FirstNumber + SecondNumber;
    // Указывает на успешное выполнение сценария действия.
    return true;
}
```

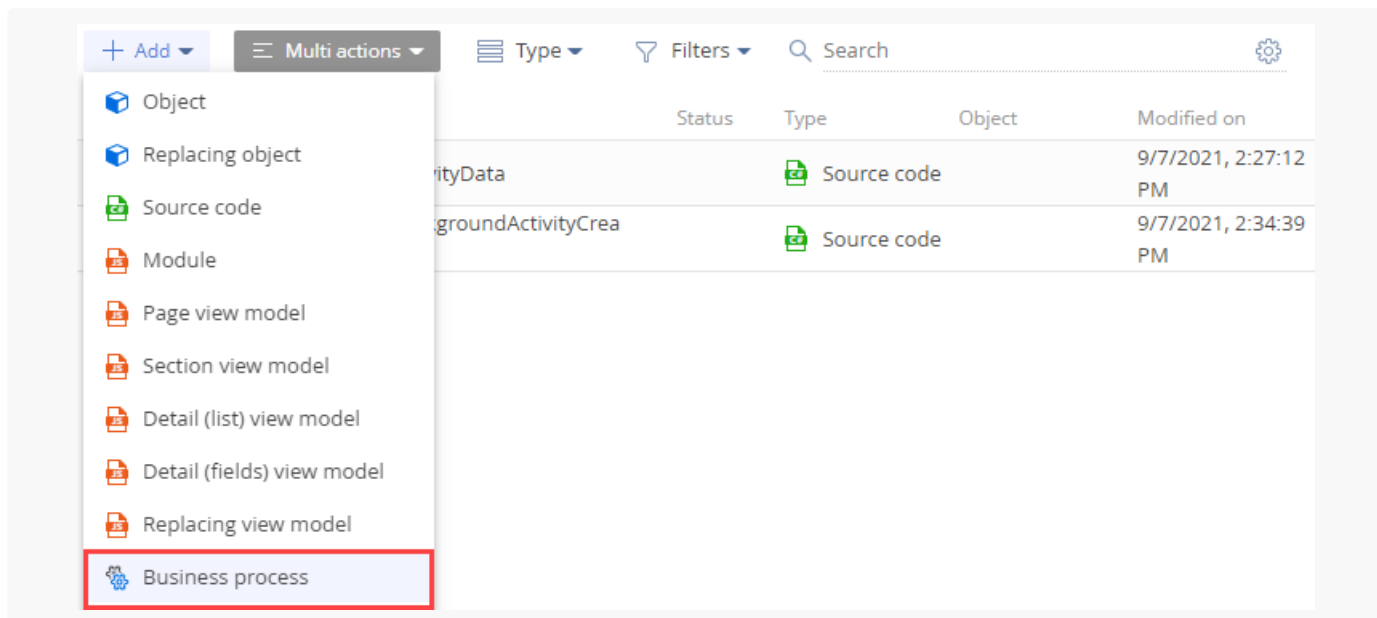
2. На панели инструментов дизайнера нажмите [ Опубликовать ] ([ Publish ]).

После успешной публикации схемы пользовательское действие процесса можно использовать при создании бизнес-процессов.

## 4. Выполнить тестирование

Для проверки работоспособности созданного действия процесса создайте новый бизнес-процесс. Подробно о том, как создать пользовательский бизнес-процесс, можно узнать из статьи [Дизайнер процессов](#).

1. [Перейдите в раздел \[ Конфигурация \]](#) ([ Configuration ]) и выберите пользовательский [пакет](#), в который будет добавлена схема.
2. На панели инструментов реестра раздела нажмите [ Добавить ] —> [ Бизнес процесс ] ([ Add ] —> [ Business process ]).



3. Заполните **свойства процесса**.

- На панели настройки элементов заполните свойство [ Заголовок ] ([ Title ]) — "Testing process".
- На вкладке [ Настройки ] ([ Settings ]) панели настройки элементов заполните свойство [ Имя ] ([ Code ]) — "UsrTestingProcess".

Process

Testing process

SETTINGS PARAMETERS METHODS

Code\*

UsrTestingProcess

Version

0

Tag

Business Process

Process description

Package\*

sdkCreatingUserTask

Maximum Number of Repetitions

100

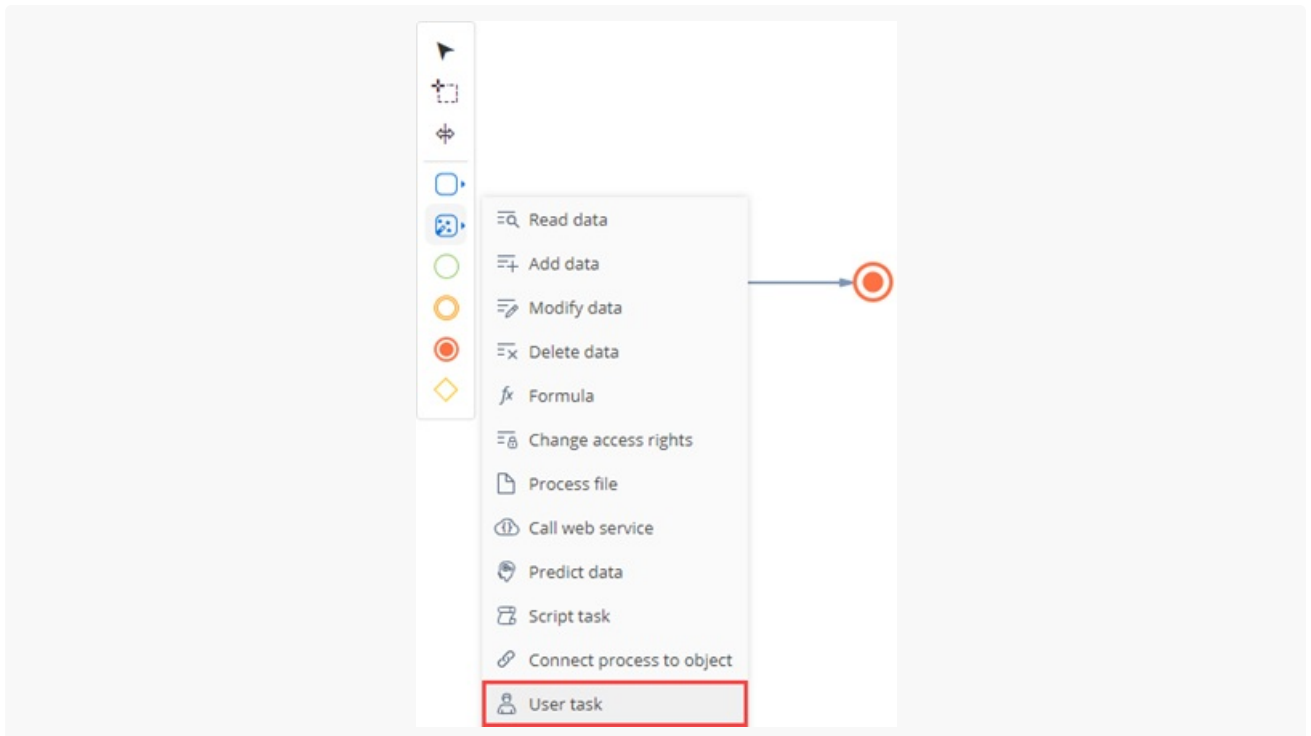
Process instance caption

[#[PropertyValue:Caption]#]

#### 4. Реализуйте бизнес-процесс.

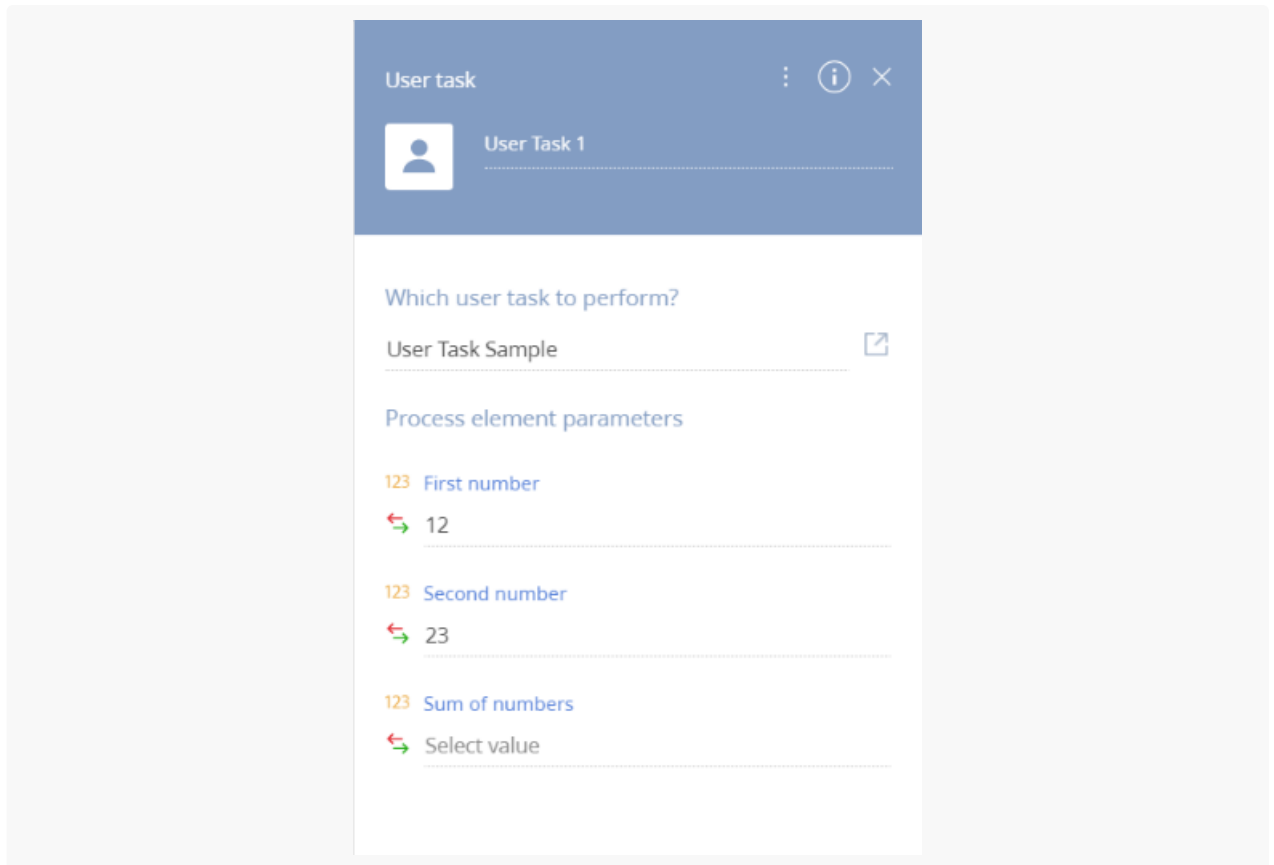
##### а. Добавьте действие процесса.

- а. В области элементов дизайнера нажмите [ *Действия системы* ] ([ *System actions* ]) и разместите элемент [ *Выполнить действие процесса* ] с в рабочей области дизайнера процессов между начальным событием [ *Простое* ] ([ *Simple* ]) и завершающим событием [ *Останов* ] ([ *Terminate* ]).



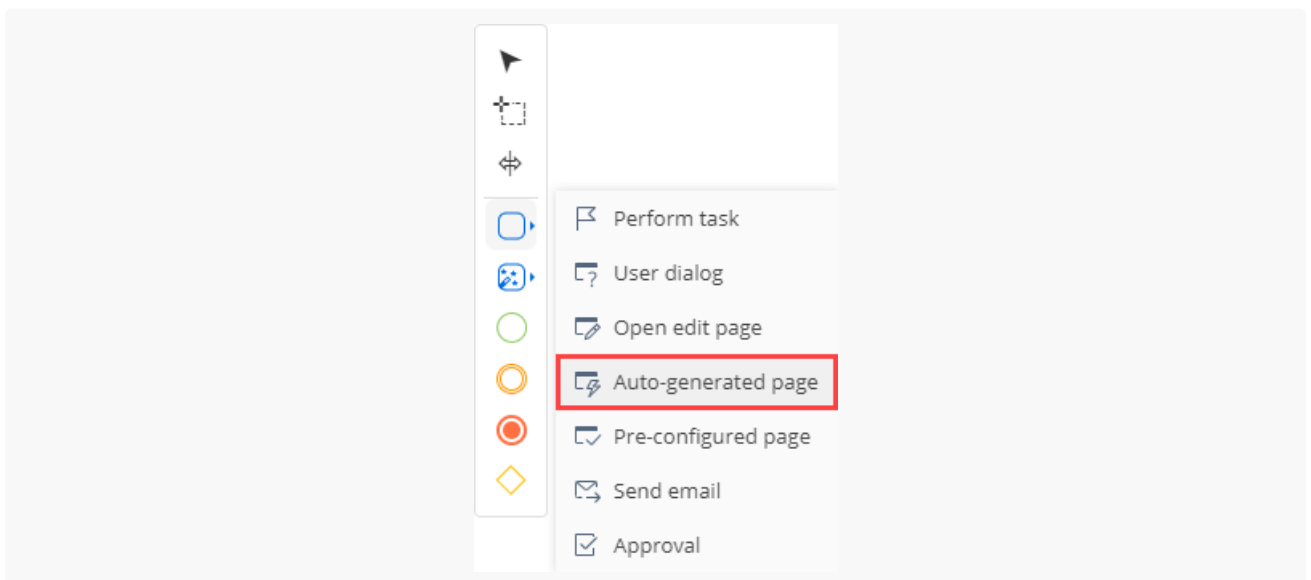
b. Заполните **свойства действия процесса**.

- [ *Какое пользовательское действие выполнить?* ] ([ *Which user task to perform?* ]) — выберите "Пример действия процесса" ("User Task Sample").
- Заполните значения параметров действия процесса.
  - [ *First Number* ] — "12".
  - [ *Second Number* ] — "23".



b. Добавьте автогенерируемую страницу.

- a. В области элементов дизайнера нажмите [ Действия пользователя ] ([ User actions ]) и разместите элемент [ Автогенерируемая страница ] ([ Auto-generated page ]) в рабочей области дизайнера процессов.




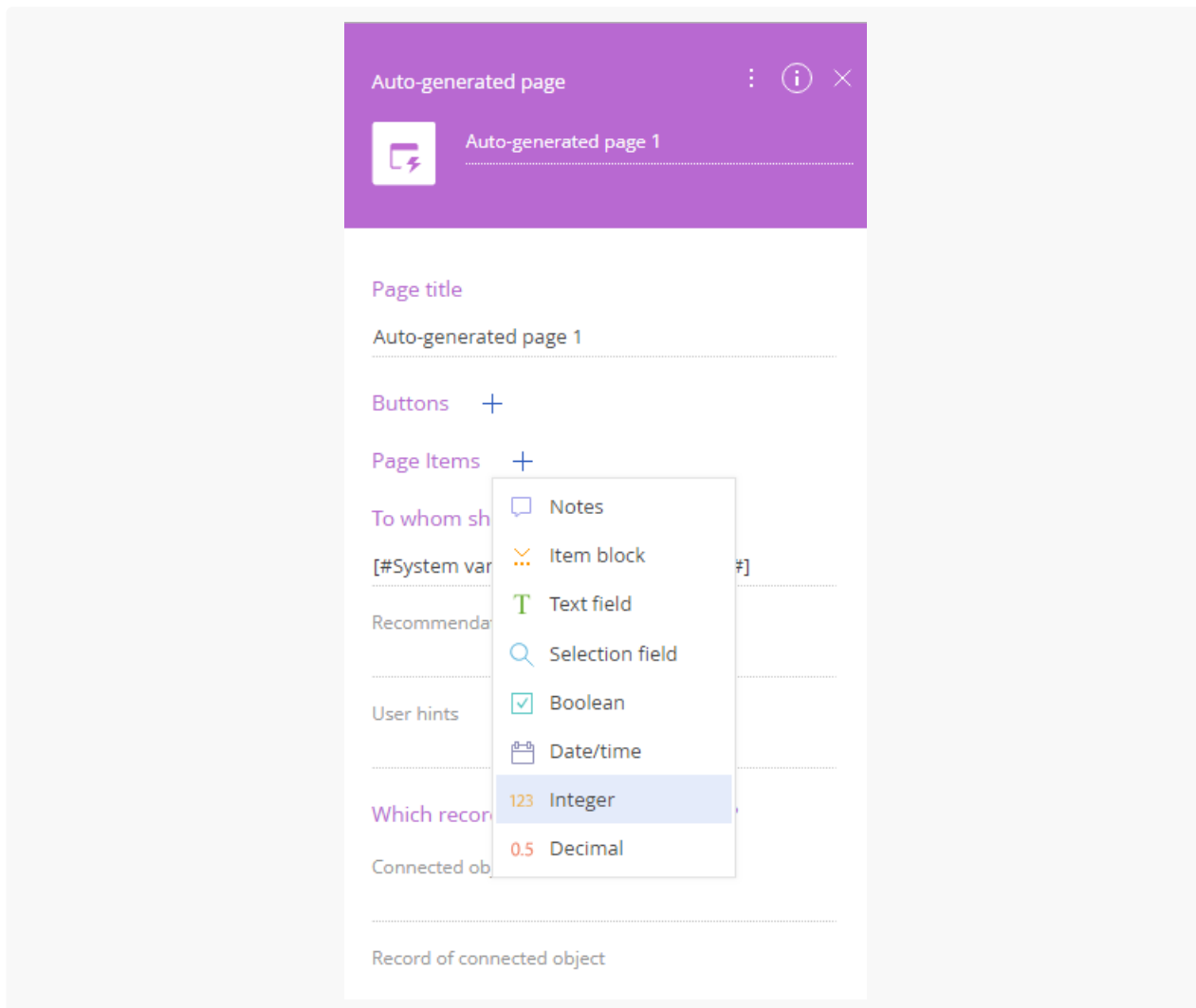
b. Заполните **свойства автогенерируемой страницы**.

- [ Заголовок ] ([ Title ]) — "Sample page".
- [ Название страницы ] ([ Page title ]) — "Sample page".




е. Добавьте элемент страницы.

- а. В блоке [ *Элементы страницы* ] ([ *Page Items* ]) нажмите кнопку  и выберите "Целое число" ("Integer").



б. Заполните **свойства элемента**.

- [ *Заголовок* ] ([ *Title* ]) — "SUM:".
- [ *Значение* ] ([ *Value* ]) — нажмите кнопку  и вызовите окно формулы значения.
- На вкладке [ *Элементы процесса* ] ([ *Process elements* ]) выберите элемент [ *Выполнить действие процесса 1* ] ([ *User task 1* ]).
- Двойным кликом выберите параметр процесса [ *Сумма чисел* ] ([ *Sum of numbers* ]) .  
В результате отобразится формула, по которой будет вычисляться отображаемое на автогенерируемой странице значение.
- Сохраните формулу.

Диалоговое окно [ *Формула* ] ([ *Formula* ])

Formula

SAVE CANCEL [Read more about formula](#)

[#User Task 1.Sum of numbers#]

< PROCESS ELEMENTS PROCESS PARAMETERS LOOKUP SYSTEM SETTINGS SYSTEM VARIABLES F>

Search process element Search element parameter

User Task 1 123 First number

Sample page 123 Second number

123 Sum of numbers

Свойства элемента страницы

Title\*

SUM:

Code\*

PageItem1

☐ Required

Value

[#User Task 1.Sum of numbers#]

SAVE CANCEL

е. Сохраните изменения.

Настройки автогенерируемой страницы представлены на рисунке ниже.

Auto-generated page

Sample page

Page title

Sample page

Buttons +

Page Items +

123 SUM:

[#User Task 1.Sum of numbers#]

To whom should the page be shown?

[#System variable.Current user contact#]

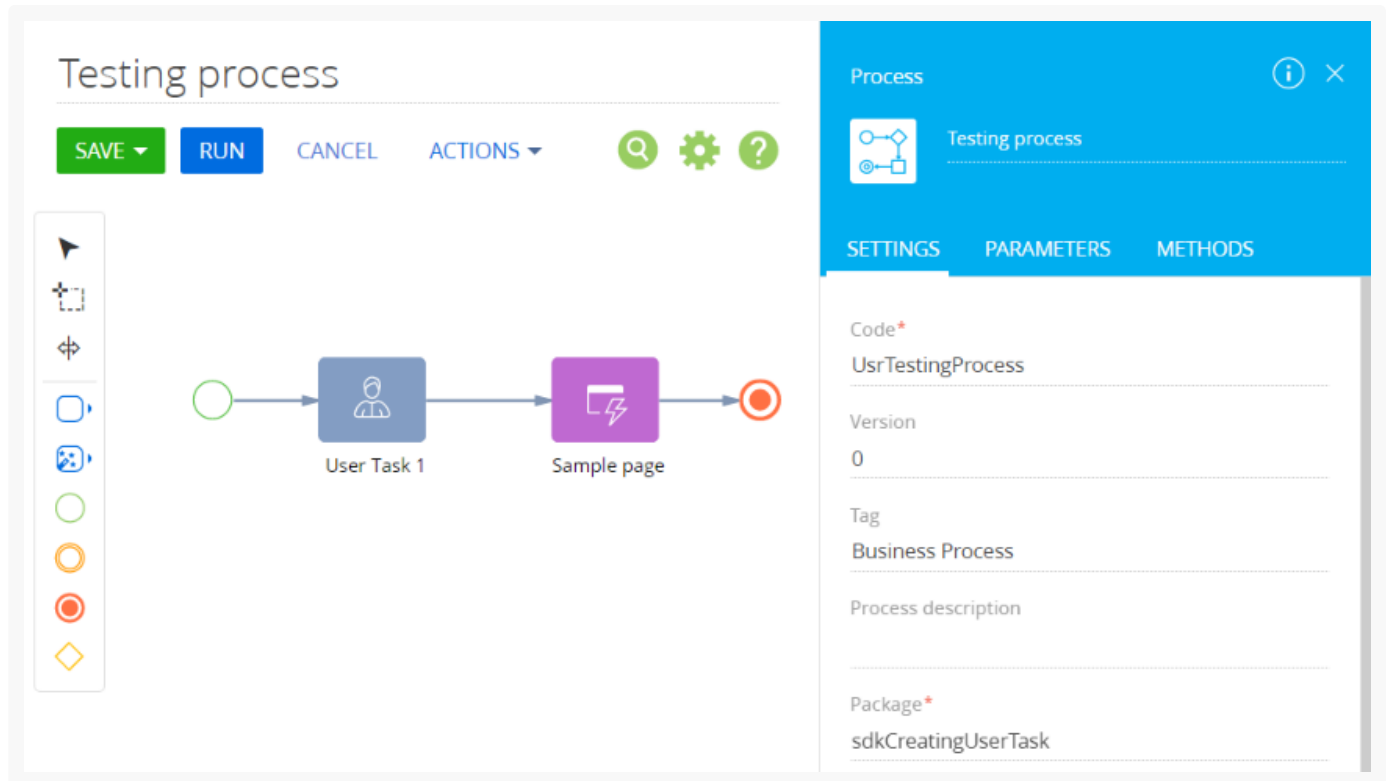
Recommendation to user

User hints

Which record to connect the page to?

Connected object

Бизнес-процесс представлен на рисунке ниже.



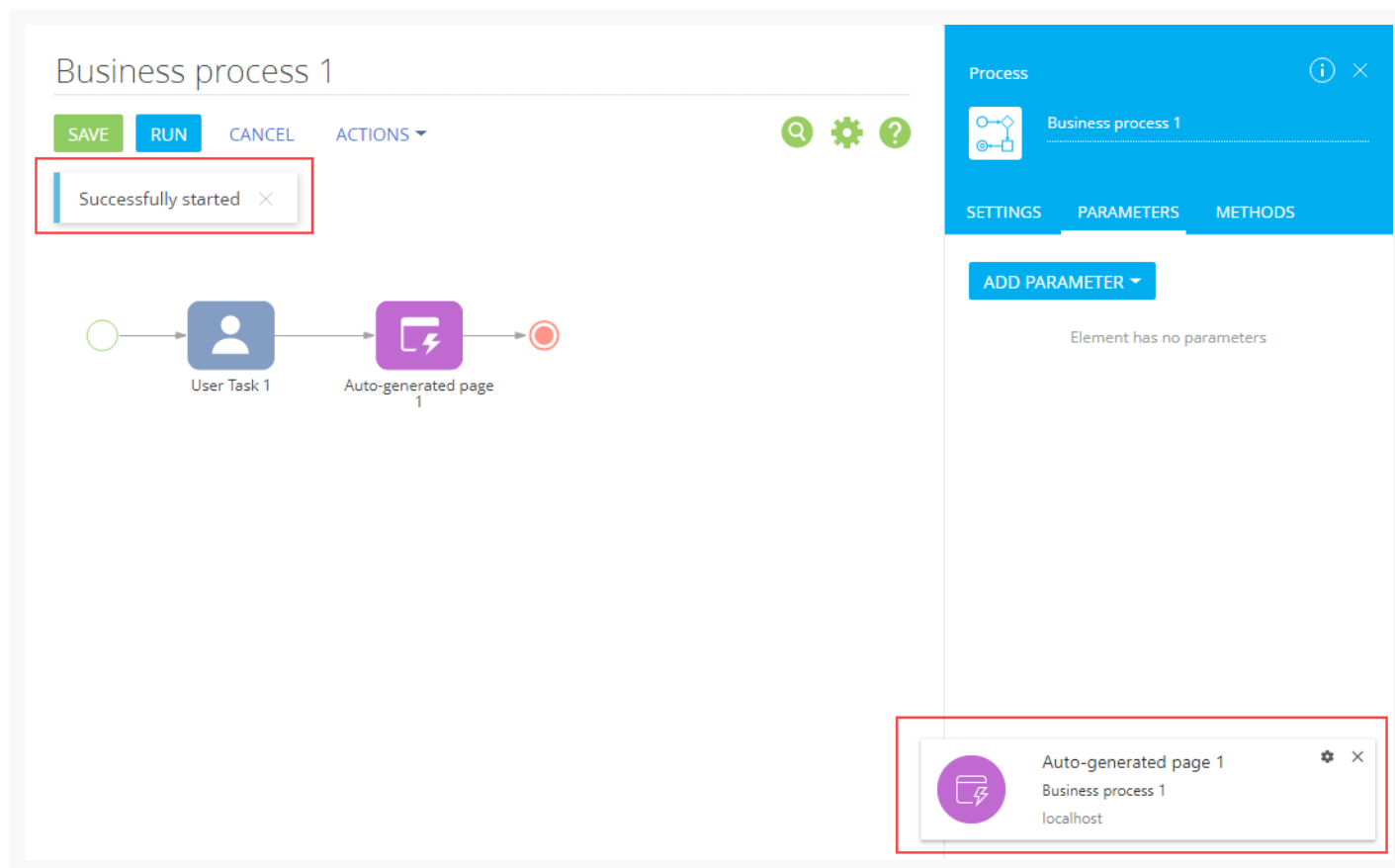
5. На панели инструментов дизайнера процессов нажмите [ Сохранить ] ([ Save ]).

6. После сохранения процесса, запустите его на выполнение.

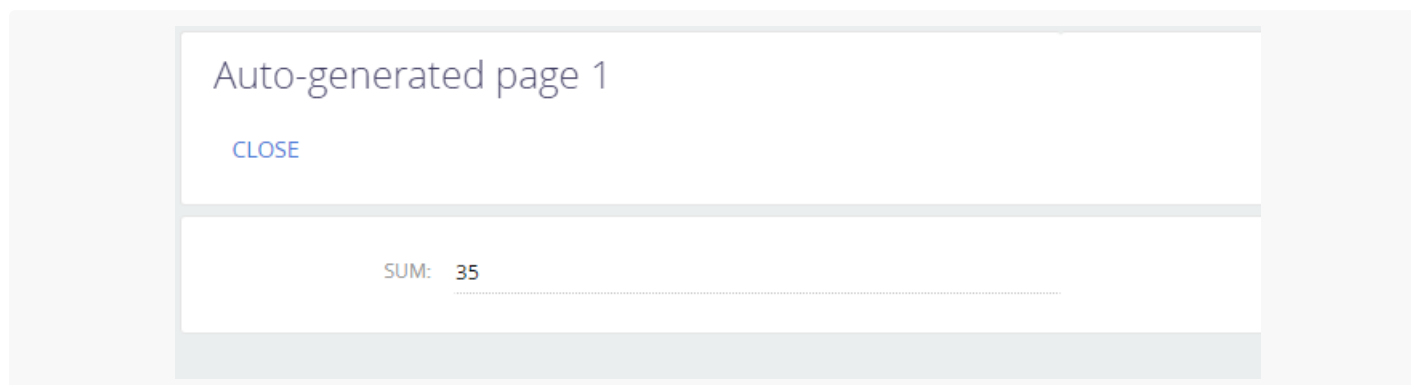
## Результат выполнения примера

После запуска бизнес процесса появится сообщение, нажав на которое можно перейти к странице отображения результатов бизнес-процесса.

Сообщение о запуске бизнес-процесса



В результате выполнения бизнес-процесса Testing process отображается страница, которая вычисляет сумму значений, заданный в качестве параметров бизнес-процесса.



## Добавить действие процесса на вкладку [Элементы процесса]

Если созданный пользовательский элемент действия процесса планируется часто использовать, то для удобства его можно добавить на вкладку Элементы процесса (Process elements) дизайнера процессов. Для этого в базе данных выполните следующий SQL-скрипт.

### SQL-скрипт

## MSSQL

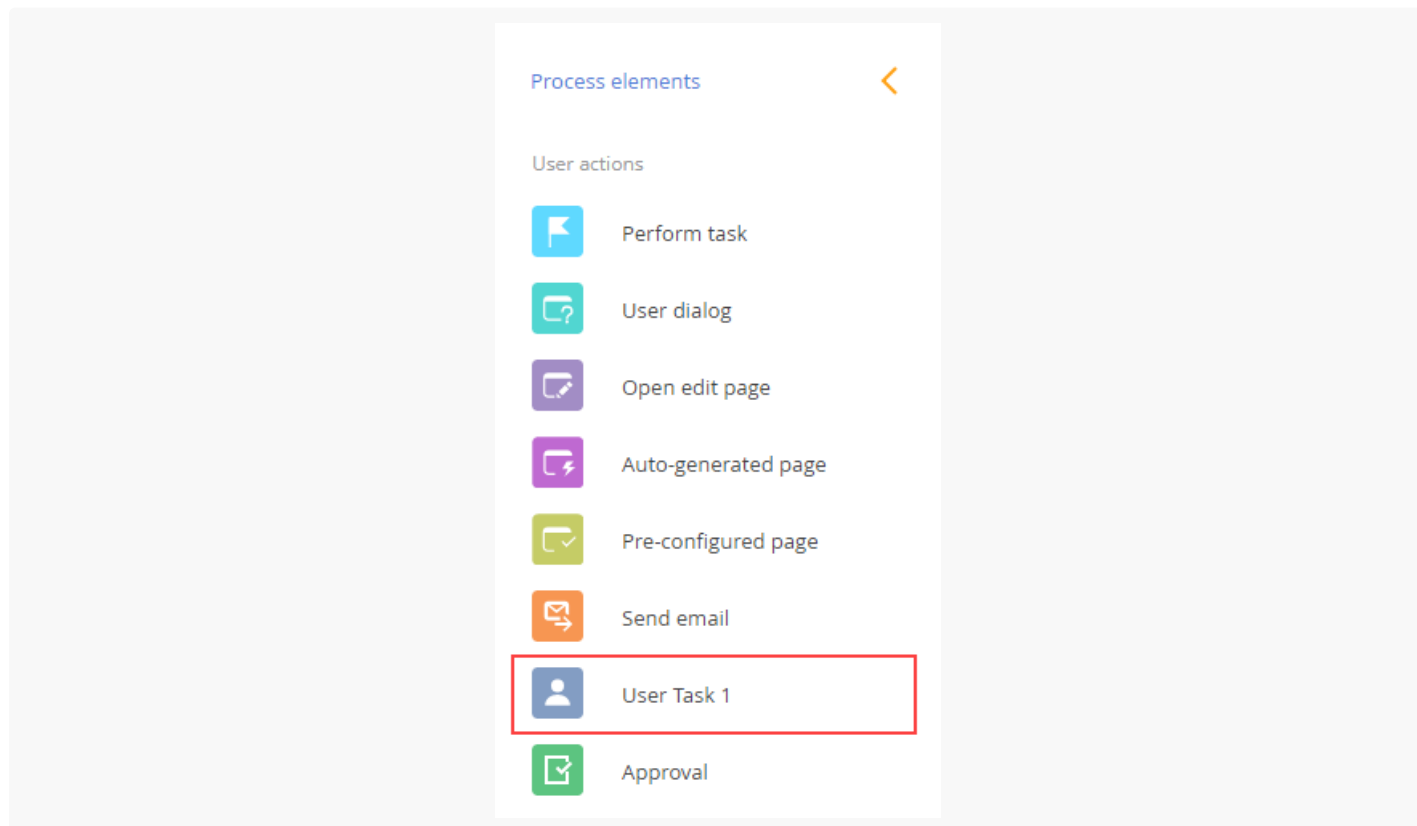
```
-- UsrSampleProcessUserTask – название схемы действия процесса.
insert into SysProcessUserTask(SysUserTaskSchemaUid, Caption)
select s.Uid, s.Caption from SysSchema s
where s.Name = 'UsrSampleProcessUserTask'
```

## PostgreSQL

```
-- UsrSampleProcessUserTask – название схемы действия процесса.
INSERT INTO SysProcessUserTask (SysUserTaskSchemaUid, Caption)
VALUES
(
    SELECT s.Uid, s.Caption FROM SysSchema AS s
    WHERE s.Name = 'UsrSampleProcessUserTask'
)
```

После перезагрузки приложения (или его компиляции) элемент отобразится на вкладке.

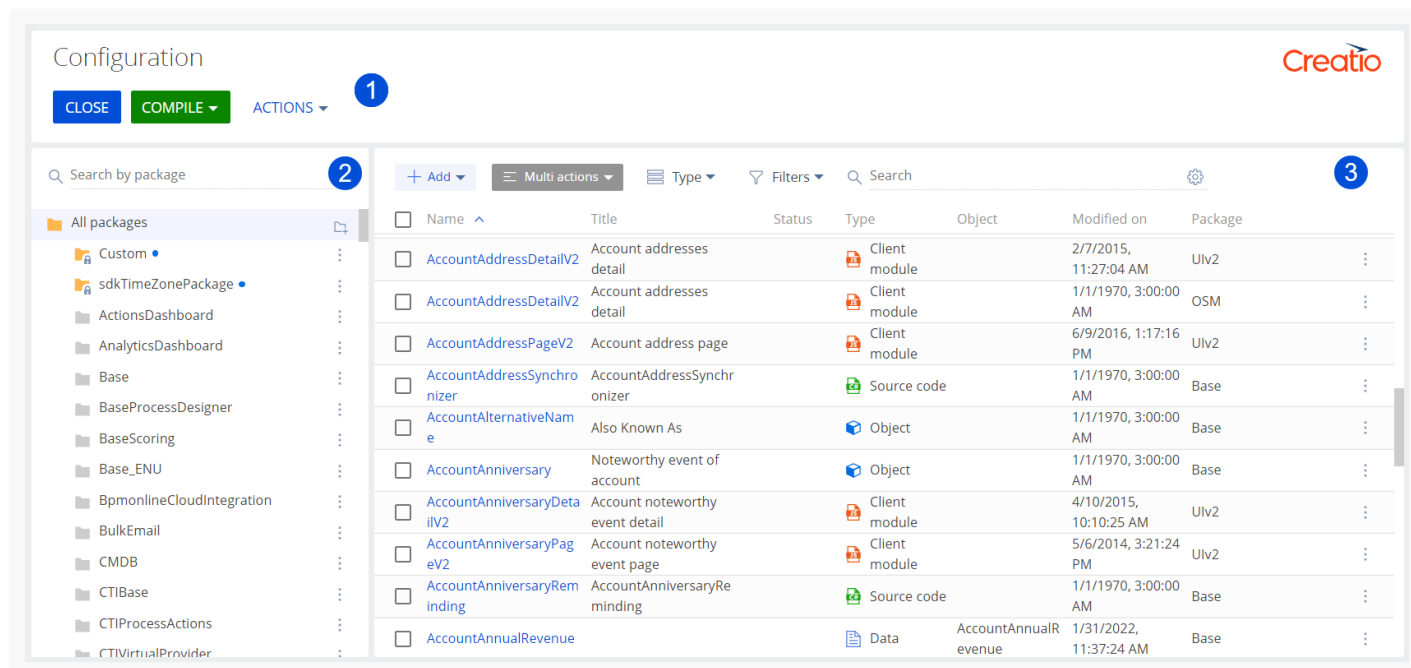
Пользовательский элемент на вкладке



# Операции в Creatio IDE

## ОСНОВЫ

Операции со структурными элементами позволяет выполнять раздел [ *Конфигурация* ] ([ *Configuration* ]), интерфейс которого представлен на рисунке ниже.



**Функциональные области** интерфейса раздела [ *Конфигурация* ] ([ *Configuration* ]):

- Панель инструментов (1).
- Область работы с пакетами (2).
- Рабочая область (3).

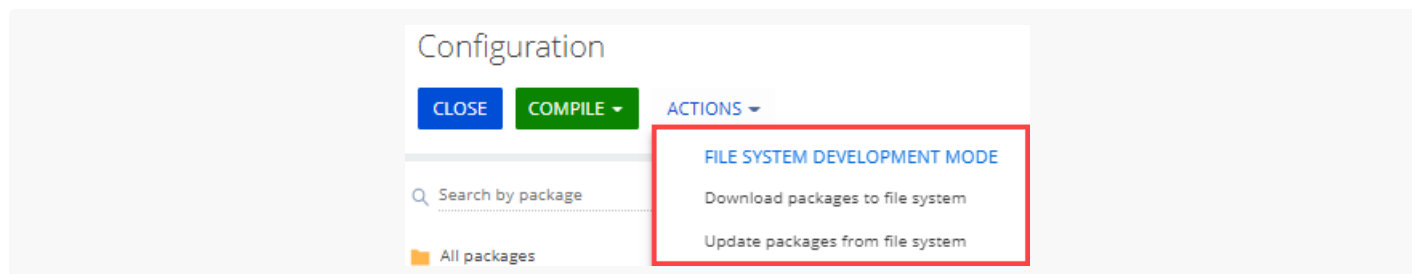
Интерфейс раздела [ *Конфигурация* ] ([ *Configuration* ]) описан в статье [Creatio IDE](#).

**Структурные элементы**, операции с которыми позволяет выполнять Creatio IDE:

- Файловая система.
- Пакеты.
- Хранилища SVN.
- Конфигурационные элементы.
- Исходный код.
- Конфигурация.

## Управлять файловой системой

Управление файловой системой реализовано в группе действий [ *Разработка в файловой системе* ] ([ *File system development mode* ]) выпадающего списка [ *Действия* ] ([ *Actions* ]) панели инструментов (1).



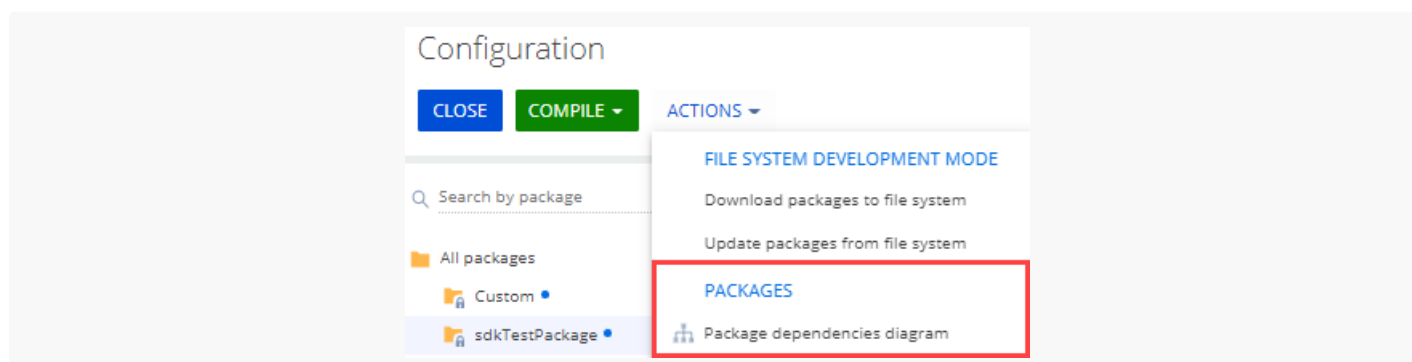
**Группа действий** [ *Разработка в файловой системе* ] ([ *File system development mode* ]) позволяет:

- Загрузить пакеты из базы данных приложения в каталог `...\Terrasoft.WebApp\Terrasoft.Configuration\Pkg` (пункт [ *Выгрузить все пакеты в файловую систему* ] ([ *Download packages to file system* ])).
- Загрузить пакеты из каталога `...\Terrasoft.WebApp\Terrasoft.Configuration\Pkg` в базу данных приложения (пункт [ *Обновить пакеты из файловой системы* ] ([ *Update packages from file system* ])).

Пункты группы действий [ *Разработка в файловой системе* ] ([ *File system development mode* ]) активны только при включенном режиме разработки в [файловой системе](#). Подсказка по включению режима отображается при наведении курсора на любой пункт текущей группы действий. Чтобы **настроить Creatio для работы в файловой системе**, воспользуйтесь инструкцией, которая приведена в статье [Внешние IDE](#).

## Управлять пакетами

Управление пакетами реализовано в группе действий [ *Пакеты* ] ([ *Packages* ]) выпадающего списка [ *Действия* ] ([ *Actions* ]) панели инструментов (1).

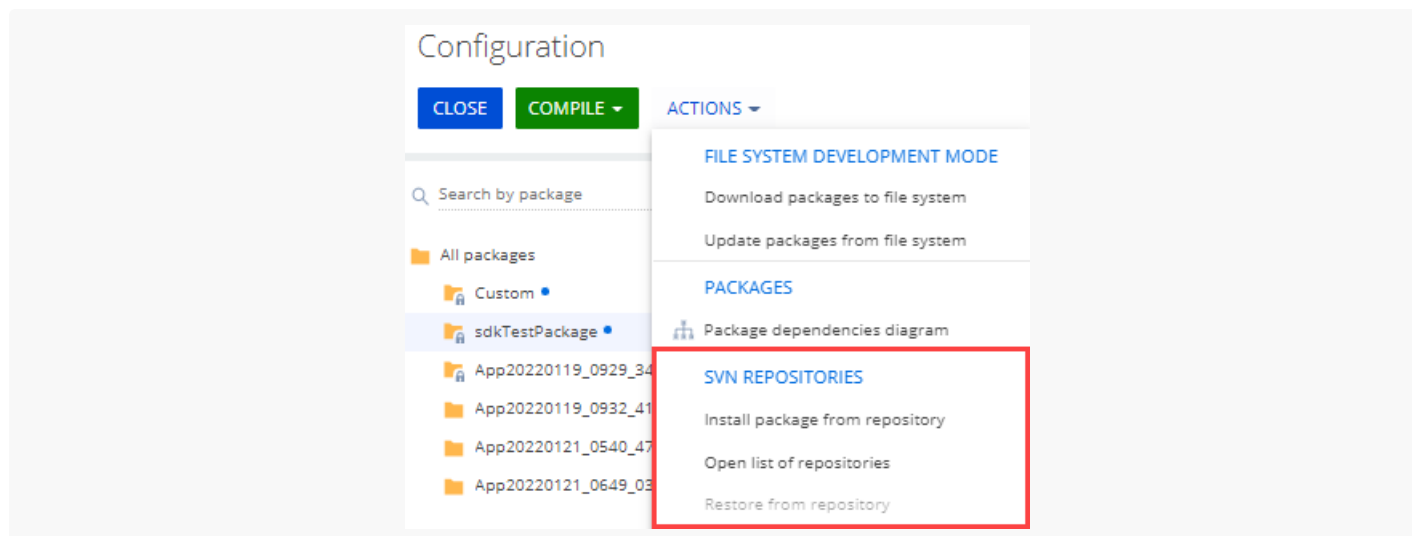


**Группа действий** [ *Пакеты* ] ([ *Packages* ]) позволяет открыть [диаграмму зависимостей](#) пакетов (пункт [ *Диаграмма зависимостей пакетов* ] ([ *Package dependencies diagram* ])).

## Управлять хранилищами SVN

Управление хранилищами SVN реализовано в группе действий [ *Хранилища SVN* ] ([ *SVN repositories* ]) выпадающего списка [ *Действия* ] ([ *Actions* ]) панели инструментов (1).





**Группа действий** [ *Хранилища SVN* ] ([ *SVN repositories* ]) позволяет:

- Установить пакет из подключенного SVN-хранилища (пункт [ *Установить пакет из хранилища* ] ([ *Install package from repository* ])).
- Открыть вкладку [ *Список хранилищ* ] ([ *List of repositories* ]) (пункт [ *Открыть список хранилищ* ] ([ *Open list of repositories* ])). Вкладка [ *Список хранилищ* ] ([ *List of repositories* ]) позволяет добавить, настроить и удалить ссылки на доступные в приложении SVN-хранилища.
- Синхронизировать конфигурацию с SVN-хранилищем до последней ревизии (пункт [ *Восстановить из хранилища* ] ([ *Restore from repository* ])). Это приведет к потере изменений, которые не зафиксированы в SVN-хранилище. Пункт неактивен, если в приложении нет пакетов, которые установлены с SVN-хранилища.

Использование системы контроля версий SVN доступно только для приложения Creatio на платформе .NET Framework. Работа с SVN-хранилищами в Creatio IDE описана в статье [Контроль версий в Creatio IDE](#).

## Управлять конфигурационными элементами


**Действия** по управлению конфигурационными элементами, которые позволяет выполнять Creatio IDE:

- Добавить конфигурационный элемент.
- Выбрать тип отображаемых конфигурационных элементов.
- Выбрать состояние отображаемых конфигурационных элементов.
- Выполнить действия с конфигурационными элементами.
- Актуализировать конфигурационные элементы.
- Выполнить поиск конфигурационного элемента.

**Реестр раздела** [ *Конфигурация* ] ([ *Configuration* ]) рабочей области (3) содержит перечень конфигурационных элементов.




**Колонки** реестра раздела:

- [ *Название* ] ([ *Name* ]) — имя конфигурационного элемента, которое было задано при его добавлении. Все конфигурационные элементы отсортированы в алфавитном порядке. Колонка позволяет

посмотреть перечень измененных конфигурационных элементов (символ  возле имени элемента). Измененные конфигурационные элементы находятся вверху перечня конфигурационных элементов реестра раздела. Подробнее о добавлении конфигурационного элемента читайте в пункте [Добавить конфигурационный элемент](#).

- [ Заголовок ] ([ Title ]) — заголовок конфигурационного элемента.
- [ Статус ] ([ Status ]) — состояние конфигурационного элемента. Выполнить фильтрацию отображаемых конфигурационных элементов по значению текущей колонки позволяет выпадающий список [ Фильтры ] ([ Filters ]) панели инструментов рабочей области раздела (3). Подробнее о состояниях конфигурационных элементов читайте в пункте [Выбрать состояние отображаемых конфигурационных элементов](#).

**Возможные значения** колонки [ Статус ] ([ Status ]):

-  — необходима актуализация конфигурационного элемента.
-  — конфигурационный элемент содержит ошибку.
-  — необходима актуализация конфигурационного элемента и при этом он содержит ошибку.

**Всплывающие подсказки**, которые позволяет увидеть Creatio IDE при наведении курсора на соответствующее значение колонки [ Статус ] ([ Status ]):

- [ Требуется установки в БД ] ([ Needs to be installed in database ]) — для конфигурационных элементов типа [ SQL сценарий ] ([ SQL script ]) и [ Данные ] ([ Data ]).
- [ Требуется обновление в БД ] ([ Needs to be updated in database ]) — для конфигурационного элемента типа [ Объект ] ([ Object ]).
- [ Требуется генерации исходного кода ] ([ Needs generate source code ]) — для конфигурационных элементов типа [ Клиентский модуль ] ([ Client module ]), [ Исходный код ] ([ Source code ]), [ Кейс ] ([ Case ]), [ Маркетинговая кампания ] ([ Marketing campaign ]), [ Бизнес-процесс ] ([ Business-process ]), [ Действие процесса ] ([ User task ]).

Подробнее о типах конфигурационных элементов читайте в пункте [Выбрать тип отображаемых конфигурационных элементов](#).

- [ Тип ] ([ Type ]) — тип конфигурационного элемента. Выполнить фильтрацию отображаемых конфигурационных элементов по значению текущей колонки позволяет выпадающий список [ Тип ] ([ Type ]) панели инструментов рабочей области раздела (3). Подробнее о типах конфигурационных элементов читайте в пункте [Выбрать тип отображаемых конфигурационных элементов](#).
- [ Объект ] ([ Object ]) — объект, с которым связаны привязанные к пакету данные. Значение колонки присутствует только для конфигурационного элемента типа [ Данные ] ([ Data ]).
- [ Дата изменения ] ([ Modified on ]) — дата изменения конфигурационного элемента.
- [ Пакет ] ([ Package ]) — название пакета, который содержит конфигурационный элемент.

Creatio IDE позволяет отсортировать данные в колонке реестра по возрастанию или по убыванию. Чтобы **отсортировать данные**, кликните по имени колонки.

**Варианты** сортировки данных в колонке реестра:

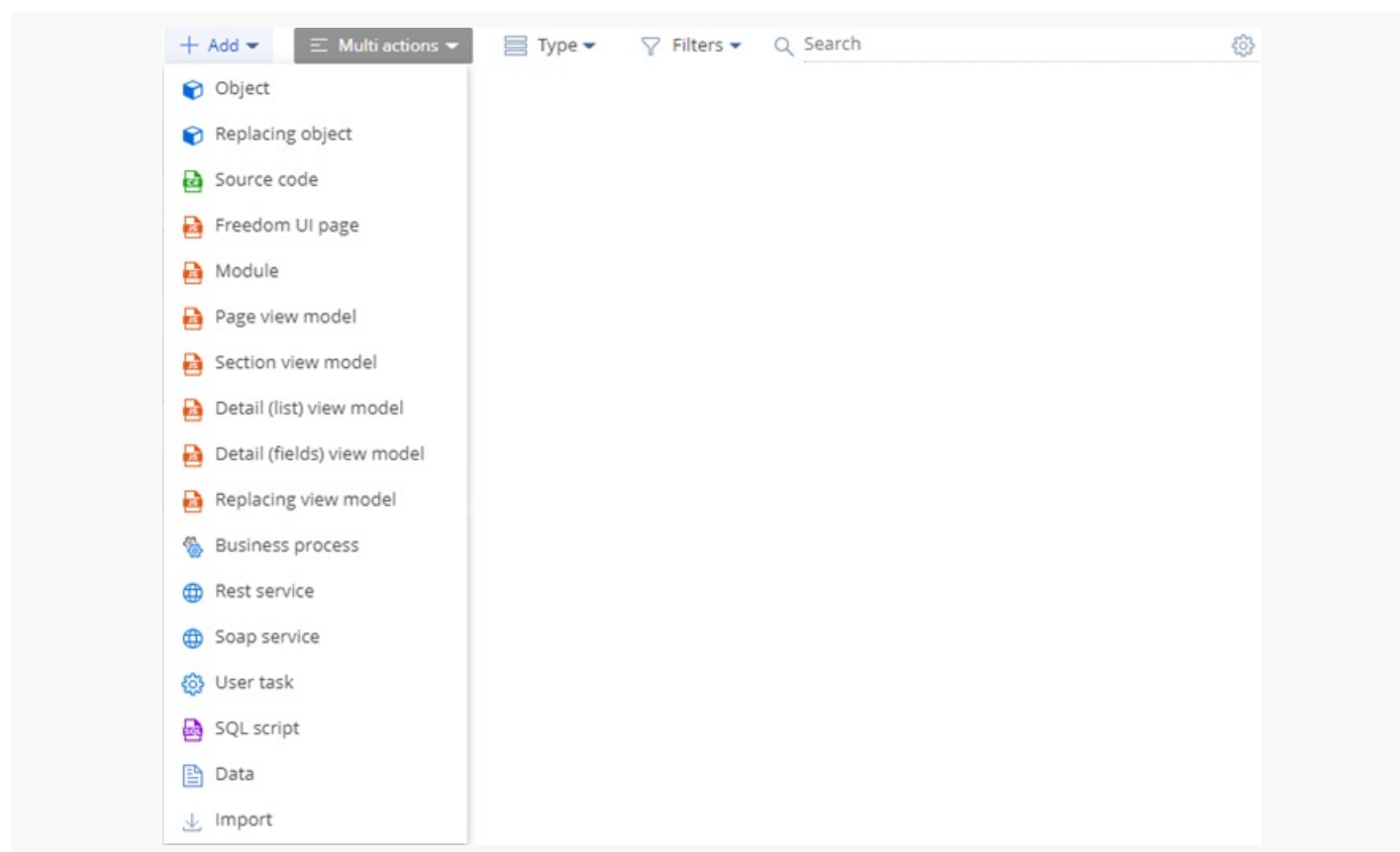
- Сортировка **в текущем пакете**. Чтобы выполнить сортировку в текущем пакете, предварительно выберите пакет в области работы с пакетами (2).
- Сортировка **во всех пакетах**. Чтобы выполнить сортировку во всех пакетах, предварительно

выберите группу [ *Все пакеты* ] ([ *All packages* ]) в области работы с пакетами (2).

## Добавить конфигурационный элемент

**Выпадающий список** [ *Добавить* ] ([ *Add* ]) панели инструментов рабочей области (3) позволяет выбрать вид добавляемого конфигурационного элемента. Предварительно в области работы с пакетами (2) необходимо выбрать пакет. Пункты выпадающего списка [ *Добавить* ] ([ *Add* ]) неактивны без предварительного выбора пакета. Creatio IDE не предоставляет возможность добавления конфигурационных элементов в предустановленные пакеты.

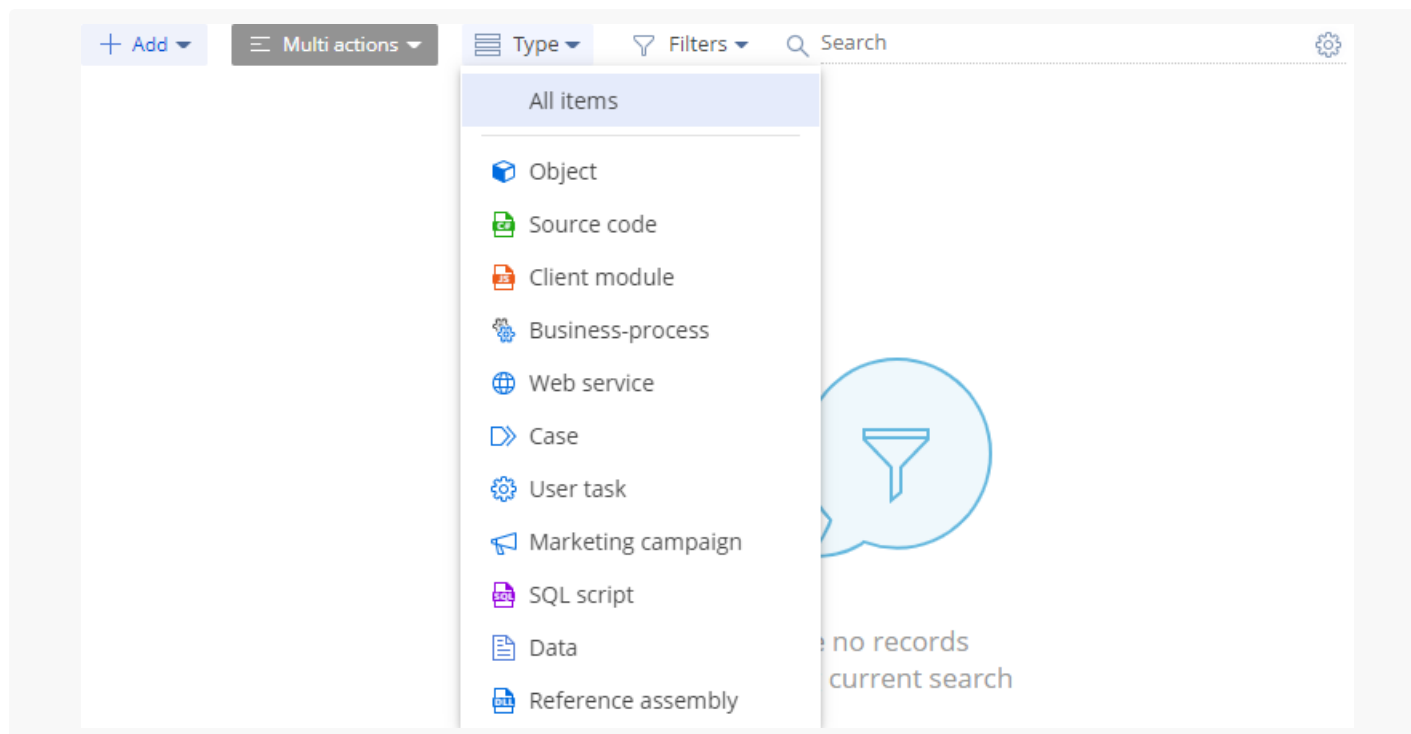
**Виды** добавляемых конфигурационных элементов на рисунке ниже.



## Выбрать тип отображаемых конфигурационных элементов

**Выпадающий список** [ *Тип* ] ([ *Type* ]) панели инструментов рабочей области (3) позволяет выбрать тип конфигурационных элементов для отображения в реестре раздела. Выбранный тип отображается в колонке [ *Тип* ] ([ *Type* ]) реестра раздела.

**Типы** отображаемых конфигурационных элементов представлены на рисунке ниже.



#### Конфигурационные элементы с типом [ Клиентский модуль ] ([ *Client module* ]):

- [ Страница Freedom UI ] ([ *Freedom UI page* ]). Доступен для версии Creatio 8.0 Atlas и выше.
- [ Модуль ] ([ *Module* ]).
- [ Модель представления страницы ] ([ *Page view model* ]).
- [ Модель представления раздела ] ([ *Section view model* ]).
- [ Модель представления детали с реестром ] ([ *Detail (list) view model* ]).
- [ Модель представления детали с полями ] ([ *Detail (fields) view model* ]).
- [ Замещающая модель представления ] ([ *Replacing view model* ]).

#### Конфигурационные элементы с типом [ Объект ] ([ *Object* ]):

- [ Объект ] ([ *Object* ]).
- [ Замещающий объект ] ([ *Replacing object* ]).

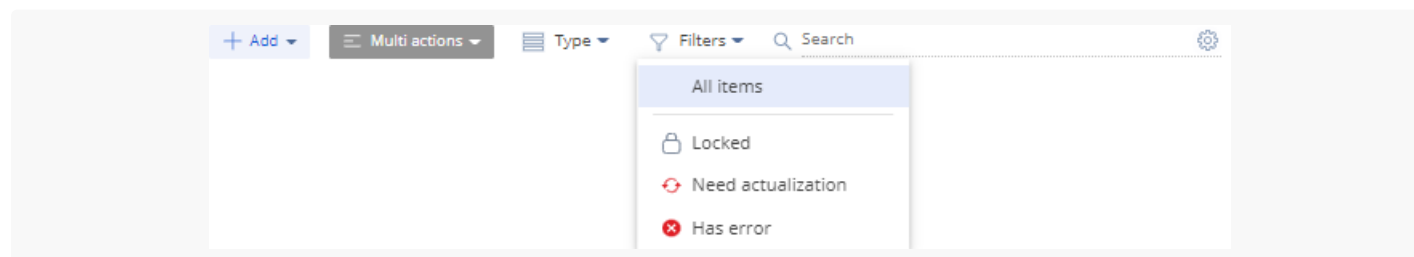
#### Конфигурационные элементы с типом [ Веб-сервис ] ([ *Web service* ]):

- [ Rest-сервис ] ([ *Rest service* ]).
- [ Soap-сервис ] ([ *Soap service* ]).

## Выбрать состояние отображаемых конфигурационных элементов

**Выпадающий список** [ Фильтры ] ([ *Filters* ]) панели инструментов рабочей области (3) позволяет выбрать состояние конфигурационных элементов для отображения в реестре раздела. Выбранный тип отображается в колонке [ Статус ] ([ *Status* ]) реестра раздела. Настройки выпадающего списка [ Фильтры ] ([ *Filters* ]) сохраняются в профиле пользователя и отображаются при входе в раздел [ Конфигурация ] ([ *Configuration* ]).

**Состояния** отображаемых конфигурационных элементов представлены на рисунке ниже.




## Выполнить действия с конфигурационными элементами

**Типы** действий, которые позволяет выполнять Creatio IDE:

- Единичные. Выполняются с одним конфигурационным элементом.
- Массовые. Выполняются с множеством конфигурационных элементов.

### Выполнить единичное действие с конфигурационным элементом

**Меню конфигурационного элемента** позволяет выполнить единичное действие с конфигурационным элементом. Чтобы **вызвать меню конфигурационного элемента**, в строке конфигурационного элемента реестра раздела нажмите на . Перечень пунктов меню конфигурационного элемента зависит от типа конфигурационного элемента. Подробнее о типах конфигурационных элементов читайте в пункте [Выбрать тип отображаемых конфигурационных элементов](#).

**Типы** единичных действий:


- Открыть объект для редактирования (пункт [ *Изменить процесс* ] ([ *Edit process* ])). Если объект создан третьими лицами, то отображается соответствующее уведомление. Отображается для конфигурационного элемента типа [ *Объект* ] ([ *Object* ]).
- Выгрузить текущий конфигурационный элемент в \*.md-формате (пункт [ *Экспортировать* ] ([ *Export* ])). Используется для экспорта текущего конфигурационного элемента с целью дальнейшей установки на другую [рабочую среду](#). Подробнее об экспорте конфигурационного элемента читайте в статье [Перенести схемы конфигурационных элементов](#).

**Конфигурационные элементы**, для которых отображается текущий пункт меню:

- [ *Объект* ] ([ *Object* ]).
- [ *Исходный код* ] ([ *Source code* ]).
- [ *Клиентский модуль* ] ([ *Client module* ]).
- [ *Бизнес-процесс* ] ([ *Business-process* ]).
- [ *Веб-сервис* ] ([ *Web service* ]).
- [ *Кейс* ] ([ *Case* ]).
- [ *Действие процесса* ] ([ *User task* ]).
- [ *Маркетинговая кампания* ] ([ *Marketing campaign* ]).
- Переместить текущий конфигурационный элемент в другой пакет (пункт [ *Переместить в другой пакет* ] ([ *Move to another package* ])). После выбора пункта отображается окно выбора пакета, в

который планируется переместить текущий конфигурационный элемент. При перемещении конфигурационного элемента Creatio IDE предоставляет возможность автоматической установки зависимостей. Отображается для конфигурационных элементов всех типов.

- Удалить конфигурационный элемент (пункт [ Удалить ] ([ *Delete* ])). Пункт неактивен для конфигурационных элементов предустановленных пакетов. Отображается для конфигурационных элементов всех типов.

Удалить конфигурационный элемент также позволяет реестр раздела (кнопка ). Данная кнопка появляется для конфигурационных элементов пользовательских пакетов при наведении курсора на запись реестра раздела [ Конфигурация ] ([ *Configuration* ]).

- Установить текущий конфигурационный элемент (пункт [ Установить ] ([ *Install* ])). Установка выполняется в базу данных. При некорректной установке данных для объекта колонки [ Объект ] ([ *Object* ]) описание ошибки отображается при наведении курсора на соответствующее значение колонки [ Статус ] ([ *Status* ]). Также описание ошибки отображается в свойствах конфигурационных элементов.

**Конфигурационные элементы**, для которых отображается текущий пункт меню:

- [ SQL сценарий ] ([ *SQL script* ]).
- [ Данные ] ([ *Data* ]).
- Обновить структуру базы данных для объекта (пункт [ Обновить структуру БД ] ([ *Update database structure* ])). Отображается для конфигурационного элемента типа [ Объект ] ([ *Object* ]).
- Сгенерировать исходный код текущего конфигурационного элемента (пункт [ Сгенерировать исходный код ] ([ *Generate source code* ])). Выполняется, если процесс содержит компилируемые элементы.

**Конфигурационные элементы**, для которых отображается текущий пункт меню:

- [ Объект ] ([ *Object* ]).
- [ Бизнес-процесс ] ([ *Business-process* ]).
- [ Действие процесса ] ([ *User task* ]).
- Открыть вкладку метаданных текущего конфигурационного элемента (пункт [ Открыть метаданные ] ([ *Open metadata* ])).

**Конфигурационные элементы**, для которых отображается текущий пункт меню:

- [ Объект ] ([ *Object* ]).
- [ Исходный код ] ([ *Source code* ]).
- [ Клиентский модуль ] ([ *Client module* ]).
- [ Бизнес-процесс ] ([ *Business-process* ]).
- [ Веб-сервис ] ([ *Web service* ]).
- [ Кейс ] ([ *Case* ]).
- [ Действие процесса ] ([ *User task* ]).
- [ Маркетинговая кампания ] ([ *Marketing campaign* ]).
- Открыть вкладку исходного кода текущего конфигурационного элемента (пункт [ Открыть исходный код ] ([ *Open source code* ])).

**Конфигурационные элементы**, для которых отображается текущий пункт меню:

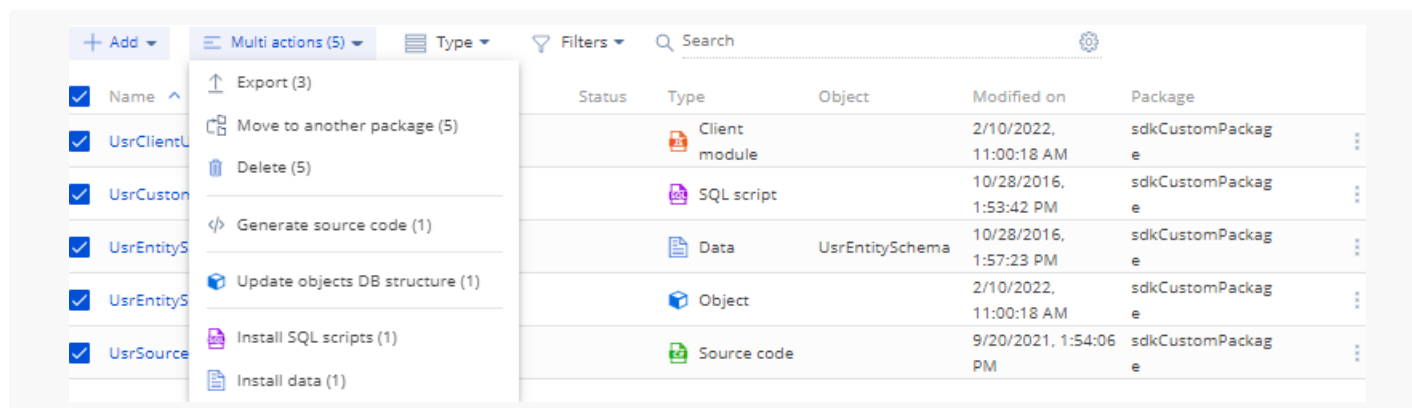
- [ *Объект* ] ([ *Object* ]).
- [ *Бизнес-процесс* ] ([ *Business-process* ]).
- [ *Действие процесса* ] ([ *User task* ]).
- Заблокировать от изменений текущий конфигурационный элемент в подключенном SVN-хранилище (пункт [ *Заблокировать элемент в SVN* ] ([ *Lock element in SVN* ])). Пункт активен только для конфигурационных элементов установленных с SVN-хранилища пакетов. Отображается для конфигурационных элементов всех типов.
- Разблокировать для изменений текущий конфигурационный элемент в подключенном SVN-хранилище (пункт [ *Разблокировать элемент в SVN* ] ([ *Unlock element in SVN* ])). Пункт активен только для конфигурационных элементов установленных с SVN-хранилища пакетов. Отображается для конфигурационных элементов всех типов.
- Отменить внесенные изменения в подключенном хранилище системы контроля версий (пункт [ *Отменить изменения* ] ([ *Discard changes* ])). Пункт активен, если к пакету с конфигурационным элементом подключено хранилище системы контроля версий. Отображается для конфигурационных элементов всех типов.
- Открыть окно свойств текущего конфигурационного элемента (пункт [ *Свойства* ] ([ *Properties* ])). Отображается для конфигурационных элементов всех типов.

## Выполнить массовые действия с конфигурационными элементами

**Выпадающий список** [ *Массовые действия* ] ([ *Multi actions* ]) панели инструментов рабочей области (3) позволяет выполнить массовые действия с выбранными конфигурационными элементами.

Предварительно в реестре раздела рабочей области (3) необходимо выбрать конфигурационные элементы, с которыми планируется выполнить массовые действия. Пункты выпадающего списка [ *Массовые действия* ] ([ *Multi actions* ]) неактивны без предварительного выбора конфигурационных элементов. Перечень отображаемых пунктов выпадающего списка [ *Массовые действия* ] ([ *Multi actions* ]) зависит от типа выбранных в реестре раздела конфигурационных элементов. Подробнее о типах конфигурационных элементов читайте в пункте [Выбрать тип отображаемых конфигурационных элементов](#).

**Массовые действия** с конфигурационными элементами представлены на рисунке ниже.



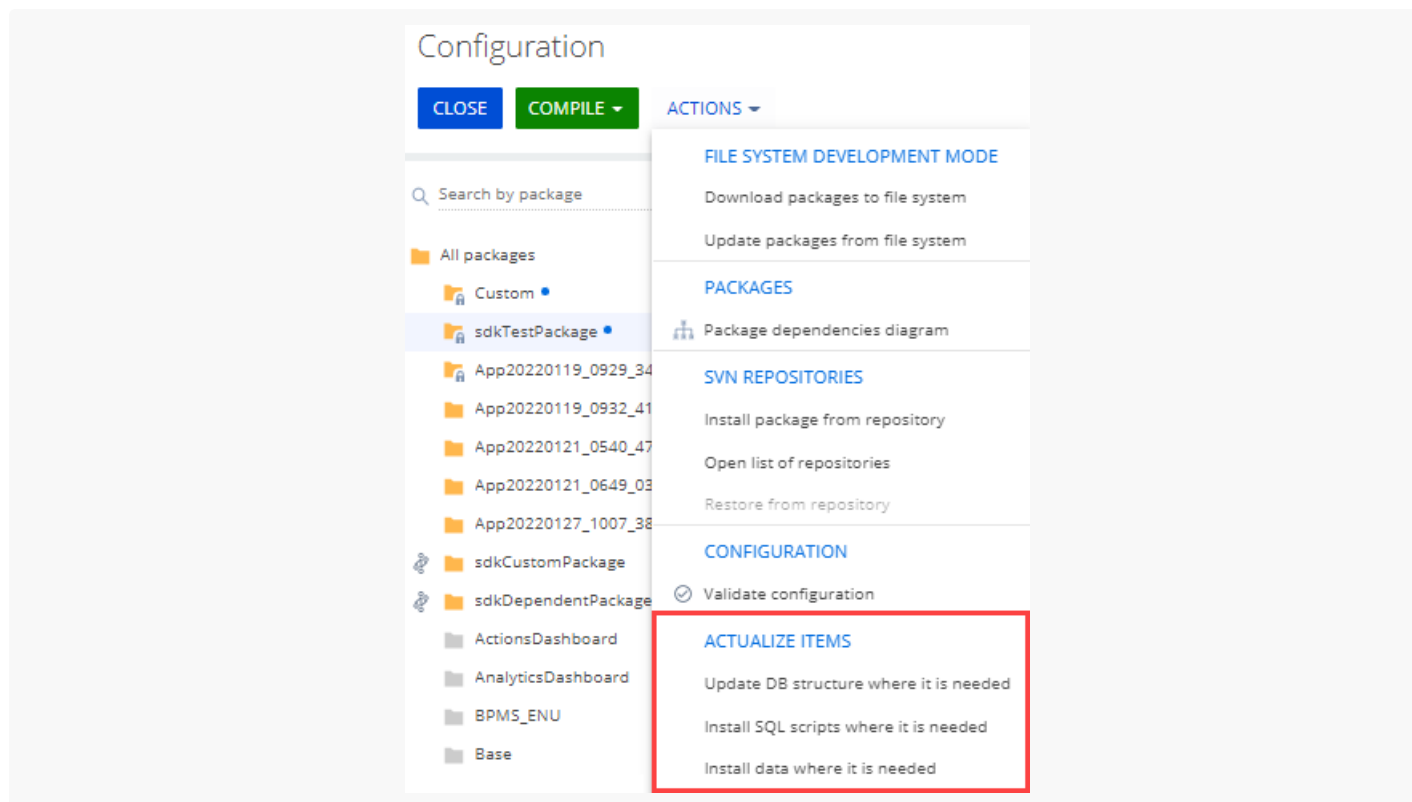
**Типы массовых действий:**

- Выгрузить выбранные конфигурационные элементы в \*.zip-архив (пункт [ *Экспортировать* ] ([ *Export* ])). Используется для экспорта выбранных конфигурационных элементов с целью дальнейшей установки на другую рабочую среду. Подробнее об экспорте пакета читайте в статье [Перенести пакеты](#).
- Переместить выбранные конфигурационные элементы пакета в другой пакет (пункт [ *Переместить в другой пакет* ] ([ *Move to another package* ])). После выбора пункта отображается окно выбора пакета, в который планируется переместить выбранные конфигурационные элементы. При перемещении конфигурационных элементов Creatio IDE предоставляет возможность автоматической установки зависимостей.
- Удалить выбранные конфигурационные элементы (пункт [ *Удалить* ] ([ *Delete* ])).
- Сгенерировать исходный код выбранных конфигурационных элементов (пункт [ *Сгенерировать исходный код* ] ([ *Generate source code* ])). Пункт отображается, если процесс содержит компилируемые элементы.
- Обновить структуру базы данных для выбранных конфигурационных элементов (пункт [ *Обновить структуру БД объектов* ] ([ *Update objects DB structure* ])).
- Установить в базу данных выбранные конфигурационные элементы типа [ *SQL сценарий* ] ([ *SQL script* ]) (пункт [ *Установить SQL сценарий* ] ([ *Install SQL script* ])).
- Установить в базу данных выбранные конфигурационные элементы типа [ *Данные* ] ([ *Data* ]) (пункт [ *Установить данные* ] ([ *Install data* ])). Устанавливаются данные для объекта колонки [ *Объект* ] ([ *Object* ]).

## Актуализировать конфигурационные элементы

Актуализация конфигурационных элементов реализована в группе действий [ *Актуализировать элементы* ] ([ *Actualize items* ]) выпадающего списка [ *Действия* ] ([ *Actions* ]) панели инструментов (1).





**Группа действий** [ *Актуализировать элементы* ] ([ *Actualize items* ]) позволяет:

- Обновить структуру базы данных для конфигурационных элементов, которые требуют обновления (пункт [ *Обновить структуру БД для требующих обновления* ] ([ *Update DB structure where it is needed* ])).
- Установить конфигурационные элементы типа [ *SQL сценарий* ] ([ *SQL script* ]), которые требуют установки (пункт [ *Установить SQL сценарии для требующих установки* ] ([ *Install SQL scripts where it is needed* ])).
- Установить конфигурационные элементы типа [ *Данные* ] ([ *Data* ]), которые требуют установки (пункт [ *Установить данные для требующих установки* ] ([ *Install data where it is needed* ])).

В результате актуализации конфигурационных элементов отобразится соответствующее уведомление.


## Выполнить поиск конфигурационного элемента

**Строка поиска** [ *Поиск* ] ([ *Search* ]) панели инструментов рабочей области (3) позволяет выполнить поиск конфигурационного элемента по его имени в реестре раздела. Настройки поиска сохраняются в профиле пользователя и отображаются при входе в раздел [ *Конфигурация* ] ([ *Configuration* ]).

**Варианты** поиска конфигурационного элемента:

- Поиск конфигурационного элемента **в текущем пакете**. Чтобы выполнить поиск в текущем пакете, предварительно выберите пакет в области работы с пакетами (2).
- Поиск конфигурационного элемента **во всех пакетах**. Чтобы выполнить поиск во всех пакетах, предварительно выберите группу [ *Все пакеты* ] ([ *All packages* ]) в области работы с пакетами (2).

Чтобы **настроить дополнительные параметры поиска**, в строке поиска [ *Поиск* ] ([ *Search* ]) панели

инструментов рабочей области (3) нажмите на .

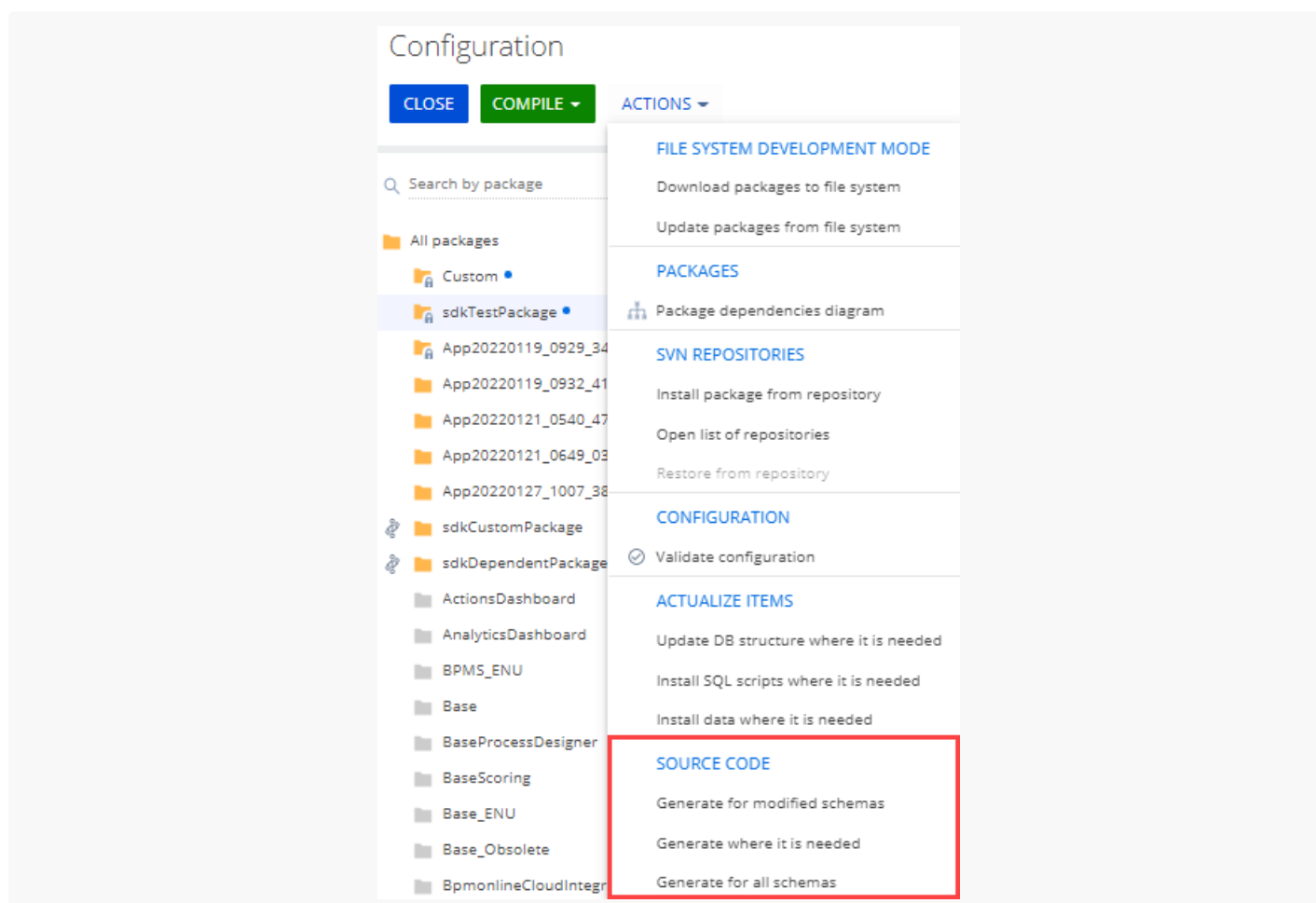
Дополнительные **параметры поиска**, которые позволяет настроить Creatio IDE:

- Поиск по заголовку (опция [ *Поиск по полю "Заголовок"* ] ([ *Search by column "Title"* ])). По умолчанию — включена.
- Поиск по уникальному идентификатору (опция [ *Поиск по полю "UID"* ] ([ *Search by column "UID"* ])).
- **Режим поиска.**
  - Название конфигурационного элемента начинается с текста, который введен в строке поиска [ *Поиск* ] ([ *Search* ]) (пункт [ *Начинается* ] ([ *Starts with* ])). По умолчанию — включен.
  - Название конфигурационного элемента содержит текст, который введен в строке поиска [ *Поиск* ] ([ *Search* ]) (пункт [ *Содержит* ] ([ *Contains* ])).
  - Название конфигурационного элемента соответствует тексту, который введен в строке поиска [ *Поиск* ] ([ *Search* ]) (пункт [ *Равно* ] ([ *Equals* ])).

Для применения внесенных изменений нажмите [ *Применить* ] ([ *Apply* ]).

## Управлять исходным кодом

Управление исходным кодом реализовано в группе действий [ *Исходный код* ] ([ *Source code* ]) выпадающего списка [ *Действия* ] ([ *Actions* ]) панели инструментов (1).



**Группа действий** [ *Исходный код* ] ([ *Source code* ]) позволяет:

- Генерировать исходный код для схем, которые были изменены в текущей конфигурации (пункт [ *Сгенерировать для измененных* ] ([ *Generate for modified schemas* ])).
- Генерировать исходный код для схем, которые требуют генерации исходного кода (пункт [ *Сгенерировать для требующих генерации* ] ([ *Generate where it is needed* ])).
- Генерировать исходный код для всех без исключения схем текущей конфигурации (пункт [ *Сгенерировать для всех схем* ] ([ *Generate for all schemas* ])). Эта операция может занять больше 10 минут.

Генерация исходного кода для схем выполняется в [фоновом режиме](#).

**Особенности** генерации исходного кода в фоновом режиме:

- Выполнение операции не влияет на работу основных пользовательских функций.
- Во время выполнения операции отсутствует возможность запуска генерации исходных кодов и выполнения компиляции приложения. При попытке запуска этих действий приложение отобразит уведомление о выполняющейся операции.
- Во время выполнения операции не рекомендуется выполнять работы, которые требуют компиляции приложения, например, устанавливать приложения и расширения, настраивать интерфейс и бизнес-логику.

В результате генерации исходного кода в [коммуникационной панели](#) отобразится соответствующее уведомление.

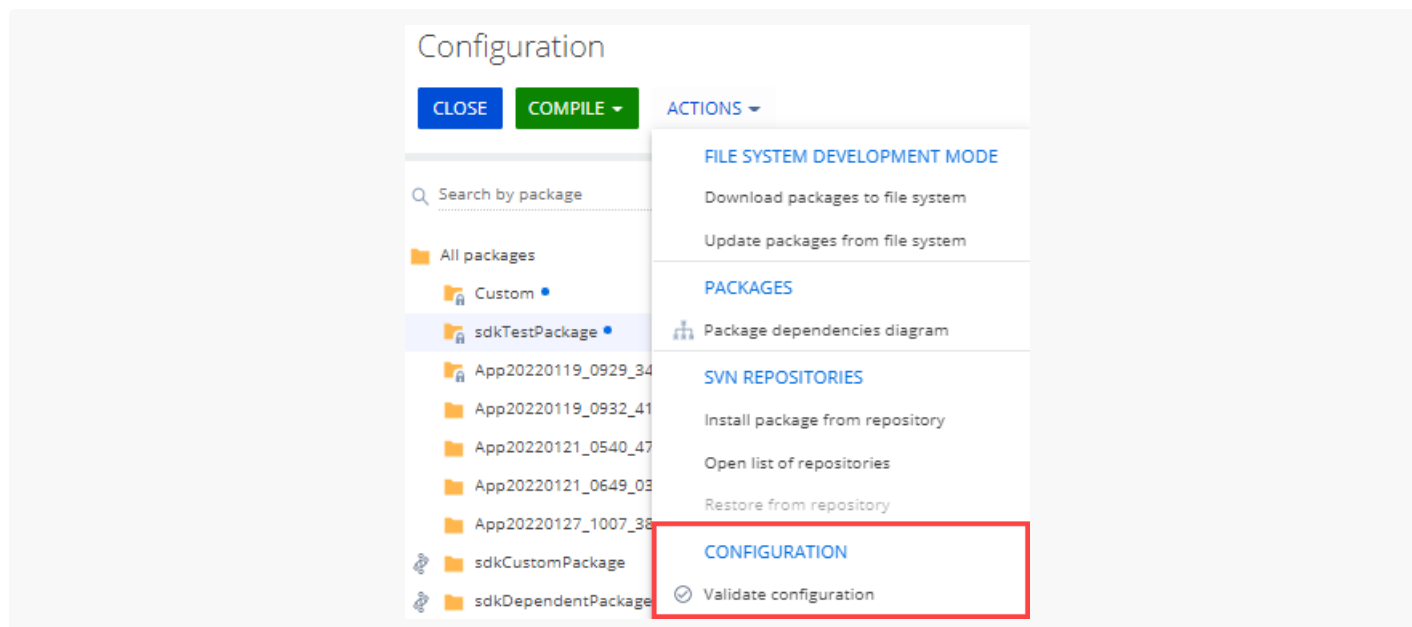
## Управлять конфигурацией

**Действия** по управлению конфигурацией, которые позволяет выполнять Creatio IDE:

- Проверить конфигурацию.
- Компилировать конфигурацию.

### Проверить конфигурацию

Проверка конфигурации реализована в группе действий [ *Проверить конфигурацию* ] ([ *Configuration* ]) выпадающего списка [ *Действия* ] ([ *Actions* ]) панели инструментов (1).



**Группа действий** [ Проверить конфигурацию ] ([ Configuration ]) позволяет проверить наличие конфликтов по зависимостям, которые могли появиться после переноса элементов между пакетами (пункт [ Проверить зависимости между элементами ] ([ Validate configuration ])).

## Компилировать конфигурацию

Действия по компиляции конфигурации выполняются на панели инструментов раздела.

**Способы** компиляции конфигурации:

- Компилировать только изменения в конфигурации (кнопка [ Компилировать ] ([ Compile ]) панели инструментов (1)).
- Компилировать конфигурацию (пункт [ Перекомпилировать все ] ([ Compile all ]) выпадающего списка кнопки [ Компилировать ] ([ Compile ]) панели инструментов (1)).



**Действия**, которые выполняются в результате компиляции конфигурации:

- Обновление исполняемых файлов.
- Выгрузка [статического контента](#) в каталог `...\Terrasoft.WebApp\conf`.

После завершения компиляции пользователь получает уведомление и изменения вступают в силу для пользователей, которые работают в текущей конфигурации.

В процессе компиляции конфигурации могут возникнуть **ошибки**, которые отображаются в диалоговом окне.

**Свойства** ошибок компиляции:

- Иконка типа ошибки (ошибка  или предупреждение .
- Имя файла с ошибкой.
- Описание ошибки.
- Код ошибки.

- Номер строки с ошибкой.

# Клиентский модуль

## Основы

**Конфигурационный элемент типа** [ *Клиентский модуль* ] ([ *Client module* ]) — это отдельный блок функциональности, который загружается и запускается по требованию. **Назначение** клиентского модуля — front-end разработка в приложении Creatio.

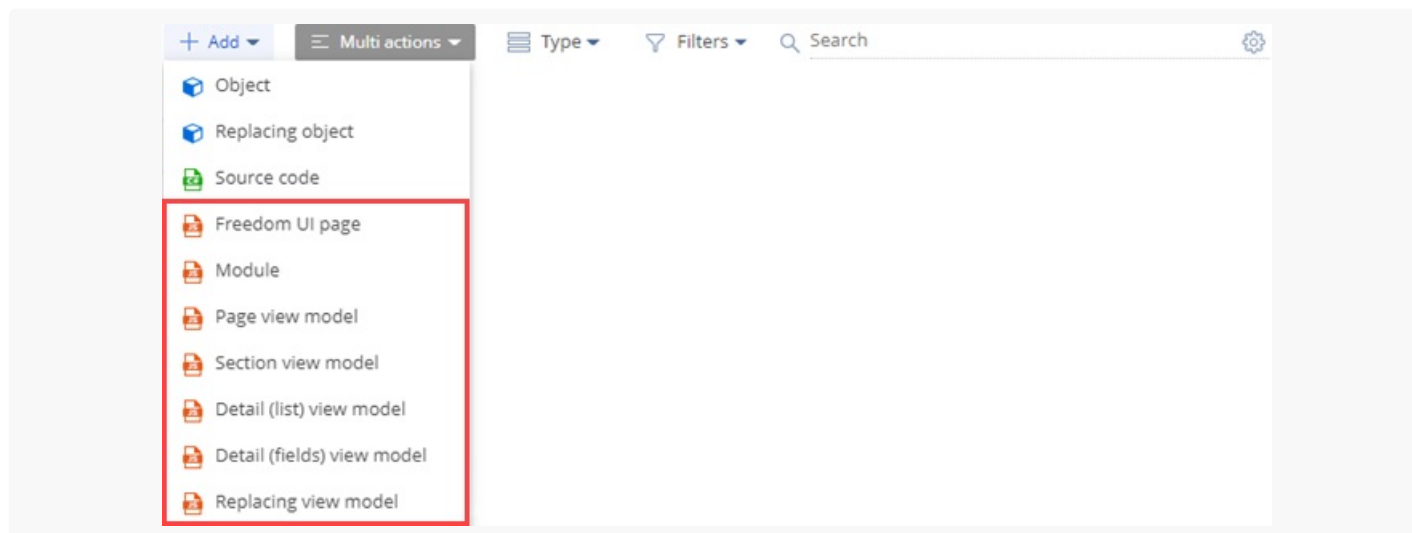
**Типы** клиентских модулей:

- Невизуальный модуль.
- Визуальный модуль.
- Замещающий модуль.

Типы клиентских модулей описаны в статье [Виды модулей](#).

Виды клиентских модулей, которые позволяет добавить Creatio IDE, представлены значениями выпадающего списка [ *Добавить* ] ([ *Add* ]) панели инструментов рабочей области раздела [ *Конфигурация* ] ([ *Configuration* ]).

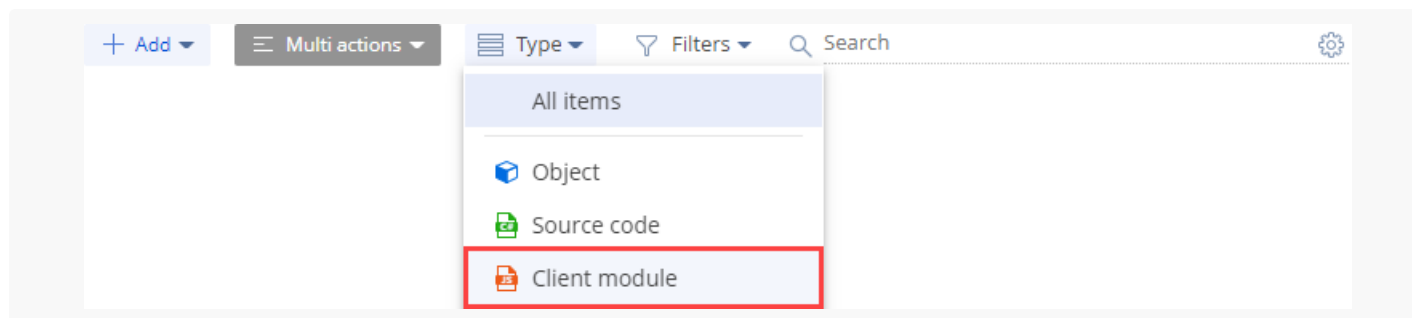
**Виды** клиентских модулей представлены на рисунке ниже.



Подробнее о видах конфигурационных элементов читайте в статье [Операции в Creatio IDE](#).

Клиентский модуль представлен схемой типа [ *Клиентский модуль* ] ([ *Client module* ]) выпадающего списка [ *Тип* ] ([ *Type* ]) панели инструментов рабочей области раздела [ *Конфигурация* ] ([ *Configuration* ]). **Схема** — основа конфигурации Creatio. С точки зрения программной реализации схема — это класс ядра, который является наследником базового класса `Schema`.

**Тип** схемы клиентского модуля представлен на рисунке ниже.



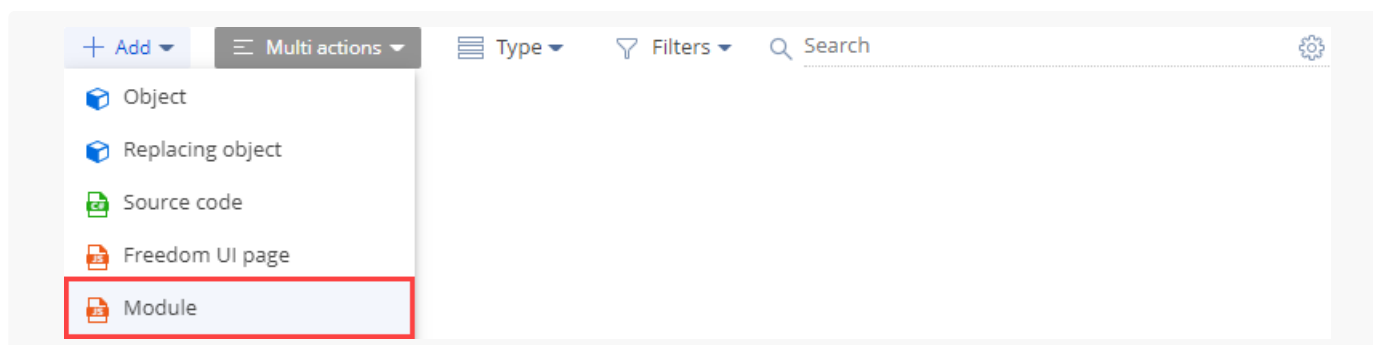
Подробнее о типах конфигурационных элементов читайте в статье [Операции в Creatio IDE](#).

## Реализовать невидуальный модуль

Невидуальный модуль представлен **видом** схемы [ *Модуль* ] ([ *Module* ]).

Чтобы **реализовать невидуальный модуль**:

1. [Перейдите в раздел \[ Конфигурация \]](#) ([ *Configuration* ]) и выберите пользовательский [пакет](#), в который будет добавлена схема.
2. На панели инструментов реестра раздела нажмите [ *Добавить* ] —> [ *Модуль* ] ([ *Add* ] —> [ *Module* ]).



3. В дизайнера модуля заполните свойства схемы.

**Основные свойства** схемы:

- [ *Код* ] ([ *Code* ]) — название схемы (обязательное свойство). Начинается с префикса (по умолчанию `usr`), который указан в системной настройке [ *Префикс названия объекта* ] ([ *Prefix for object name* ], код `SchemaNamePrefix`). Может содержать символы латинского алфавита и цифры.
- [ *Заголовок* ] ([ *Title* ]) — локализуемый заголовок схемы (обязательное свойство).
- [ *Пакет* ] ([ *Package* ]) — пользовательский пакет, в котором создается схема. Заполняется автоматически и недоступно для редактирования.
- [ *Описание* ] ([ *Description* ]) — локализуемое описание схемы.

Для применения изменений свойств нажмите [ Применить ] ([ Apply ]).

**Панель свойств** дизайнера модуля позволяет:

- Изменить основные свойства схемы (кнопка ).
- Задать дополнительные свойства схемы (кнопка ).

**Дополнительные свойства** схемы:

- [ Локализуемые строки ] ([ Localizable strings ]).
- [ Сообщения ] ([ Messages ]).
- [ Изображения ] ([ Images ]).
- [ Параметры ] ([ Parameters ]).

4. В дизайнера модуля добавьте исходный код. Название модуля в функции `define()` должно совпадать с названием схемы (свойство [ Код ] ([ Code ])).

Если при написании кода допущена ошибка, то слева возле номера строки отображается тип ошибки (ошибка или предупреждение ). При наведении курсора на тип ошибки отображается всплывающая подсказка с текстовым описанием.

5. На панели инструментов дизайнера модуля нажмите [ Сохранить ] ([ Save ]).

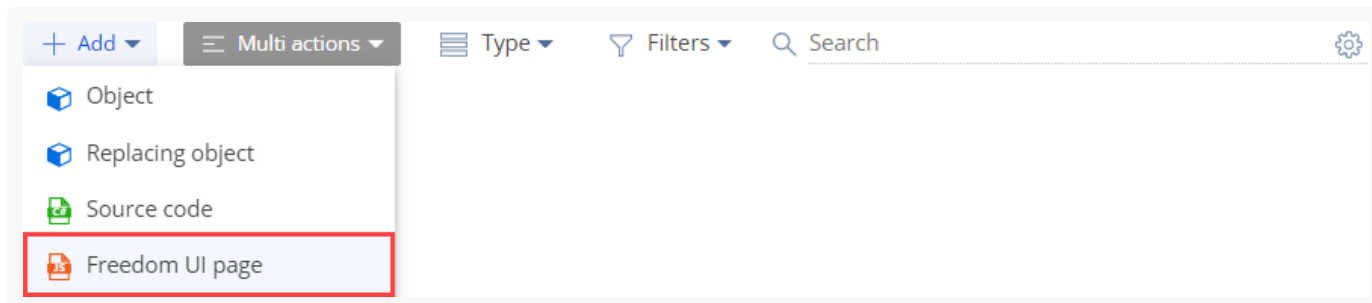
## Реализовать визуальный модуль

**Виды** визуальных модулей:

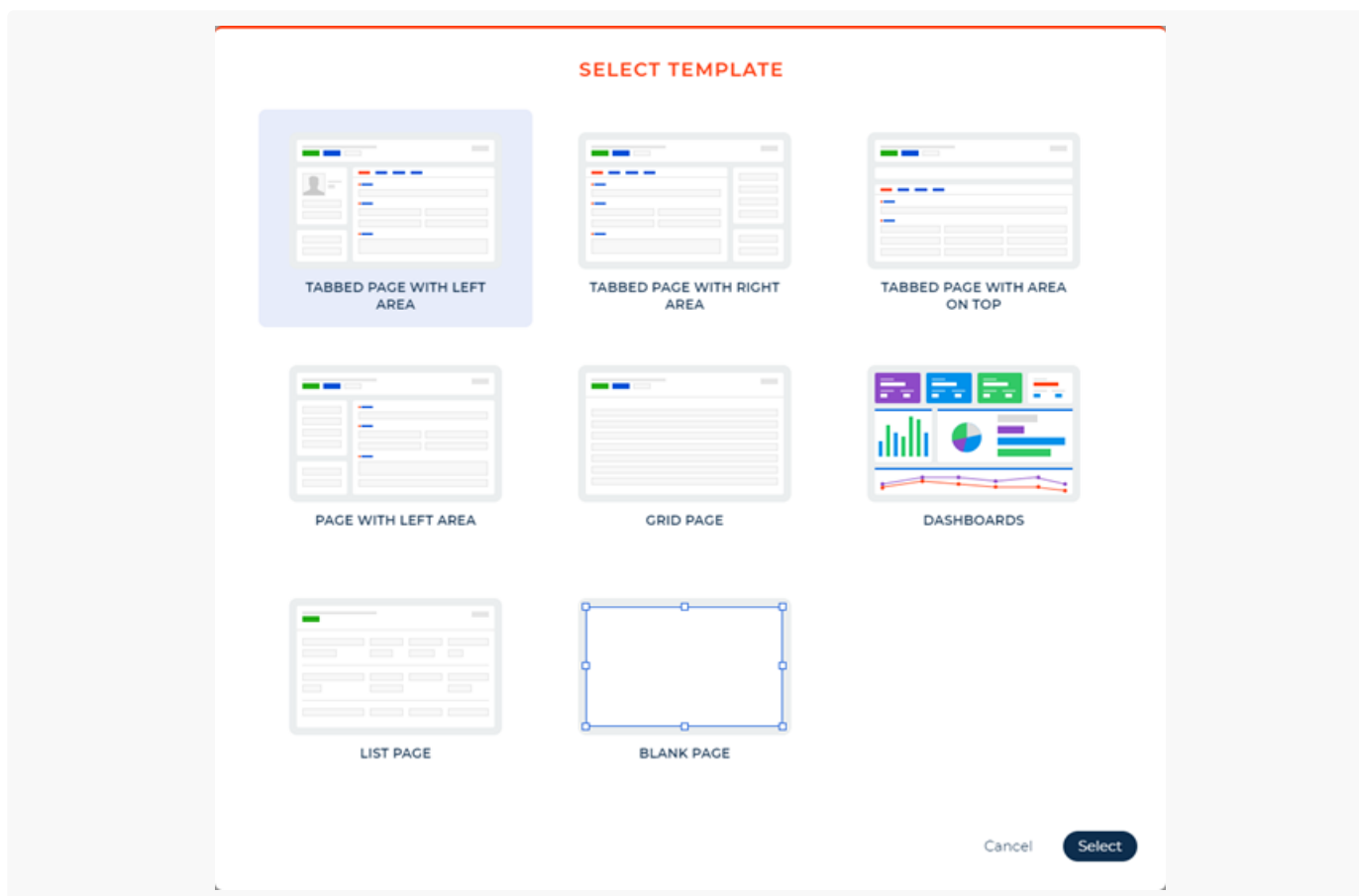
- [ Страница Freedom UI ] ([ Freedom UI page ]). Доступен для версии Creatio 8.0 Atlas и выше.
- [ Модель представления страницы ] ([ Page view model ]).
- [ Модель представления раздела ] ([ Section view model ]).
- [ Модель представления детали с реестром ] ([ Detail (list) view model ]).
- [ Модель представления детали с полями ] ([ Detail (fields) view model ]).


## Схема страницы Freedom UI

1. [Перейдите в раздел \[ Конфигурация \]](#) ([ Configuration ]) и выберите пользовательский [пакет](#), в который будет добавлена схема.
2. На панели инструментов реестра раздела нажмите [ Добавить ] —> [ Страница Freedom UI ] ([ Add ] —> [ Freedom UI page ]).



3. В зависимости от пользовательской задачи, выберите соответствующий [шаблон добавляемой страницы Freedom UI](#) и нажмите [ Выбрать ] ([ Select ]).



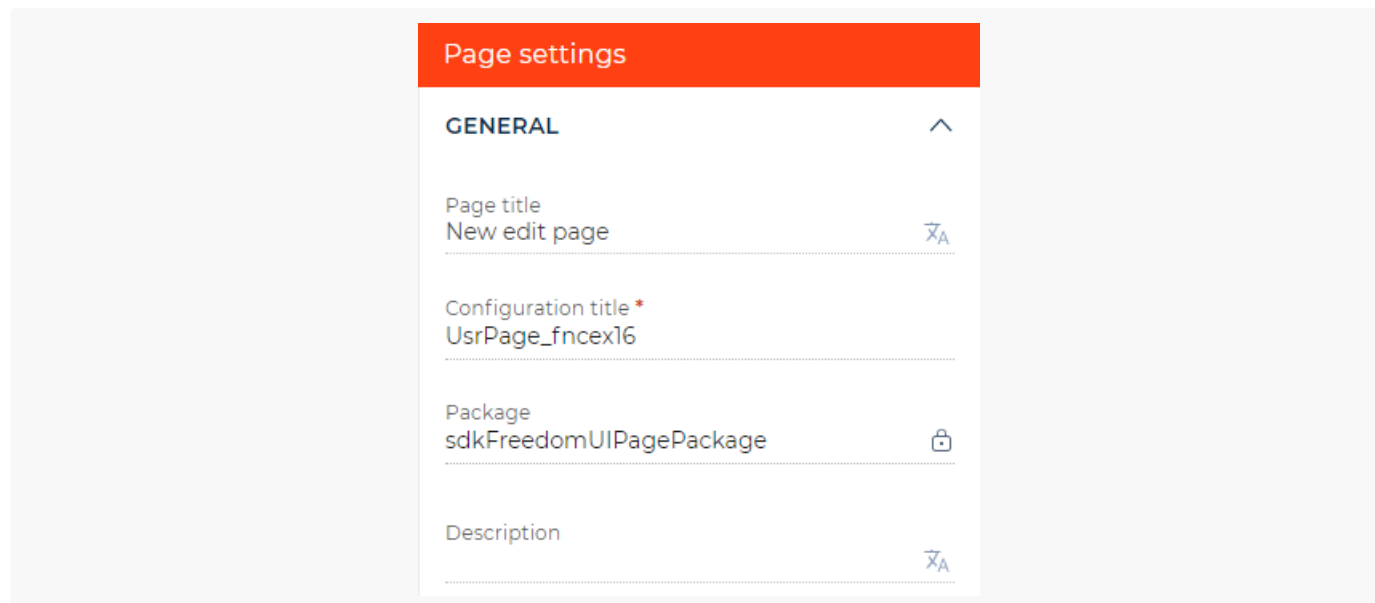
4. На панели действий Freedom UI дизайнера нажмите на кнопку  и на панели настройки заполните свойства схемы.


### Свойства схемы:

- [ Заголовок страницы ] ([ Page title ]) — локализуемый заголовок схемы.





- [ Код страницы ] ([ *Configuration title* ]) — название схемы (обязательное свойство). Начинается с префикса (по умолчанию `Usr` ), который указан в системной настройке [ Префикс названия объекта ] ([ *Prefix for object name* ], код `SchemaNamePrefix` ). Может содержать символы латинского алфавита и цифры.
- [ Пакет ] ([ *Package* ]) — пользовательский пакет, в котором создается схема. Заполняется автоматически и недоступно для редактирования.
- [ Описание ] ([ *Description* ]) — локализуемое описание схемы.



5. На панели действий Freedom UI дизайнера нажмите на кнопку . После сохранения настроек страницы открывается дизайнер модуля.

**Панель свойств** дизайнера модуля позволяет:

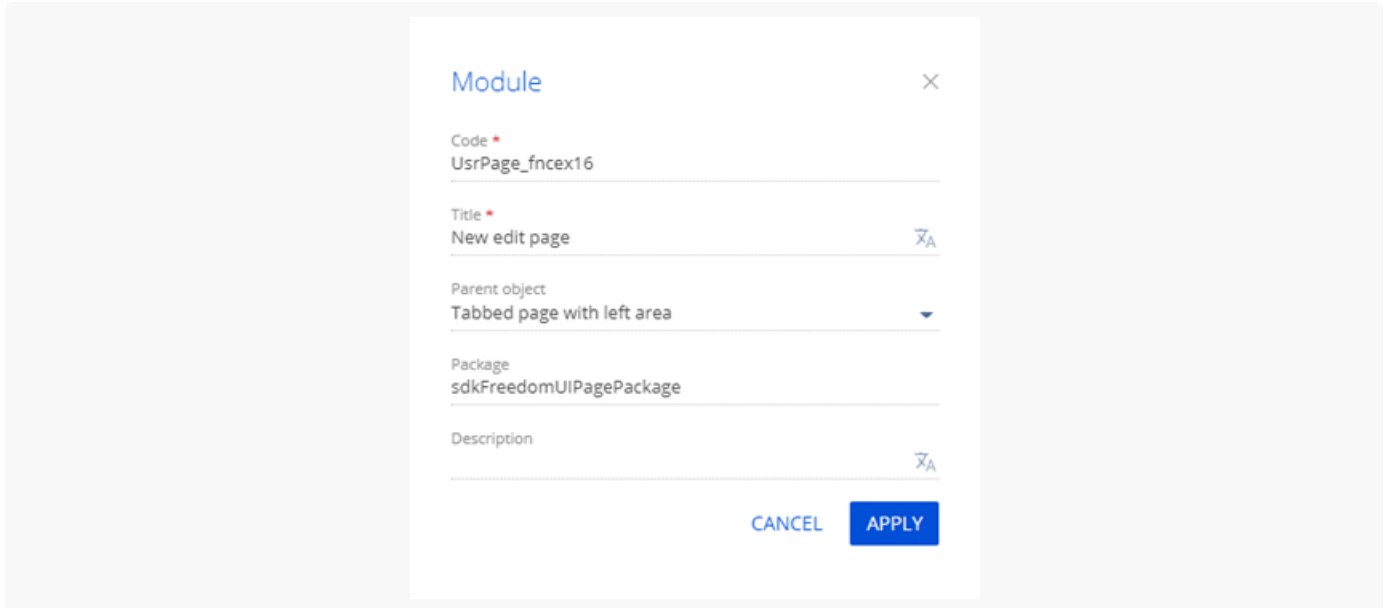
- Изменить основные свойства схемы (кнопка .
- Задать дополнительные свойства схемы (кнопка .

**Основные свойства** схемы:

- [ Код ] ([ *Code* ]) — название схемы (обязательное свойство). Идентично свойству [ Код страницы ] ([ *Configuration title* ]), которое позволяет задать Freedom UI дизайнер.
- [ Заголовок ] ([ *Title* ]) — локализуемый заголовок схемы (обязательное свойство). Идентично свойству [ Заголовок страницы ] ([ *Page title* ]), которое позволяет задать Freedom UI дизайнер.
- [ Родительский объект ] ([ *Parent object* ]) — родительский объект для текущего модуля, свойства которого необходимо наследовать. Заполняется автоматически в соответствии с шаблоном, который был выбран при добавлении страницы Freedom UI в разделе [ Конфигурация ] ([ *Configuration* ]). Доступно для редактирования.



**Важно.** Не рекомендуется изменять значение свойства [ Родительский объект ] ([ *Parent object* ]) после наполнения схемы элементами, поскольку это может привести к некорректной работе страницы.

- [ *Пакет* ] ([ *Package* ]) — пользовательский пакет, в котором создается схема. Идентично свойству [ *Пакет* ] ([ *Package* ]), которое позволяет задать Freedom UI дизайнер. Заполняется автоматически и недоступно для редактирования.
- [ *Описание* ] ([ *Description* ]) — локализуемое описание схемы. Идентично свойству [ *Описание* ] ([ *Description* ]), которое позволяет задать Freedom UI дизайнер.



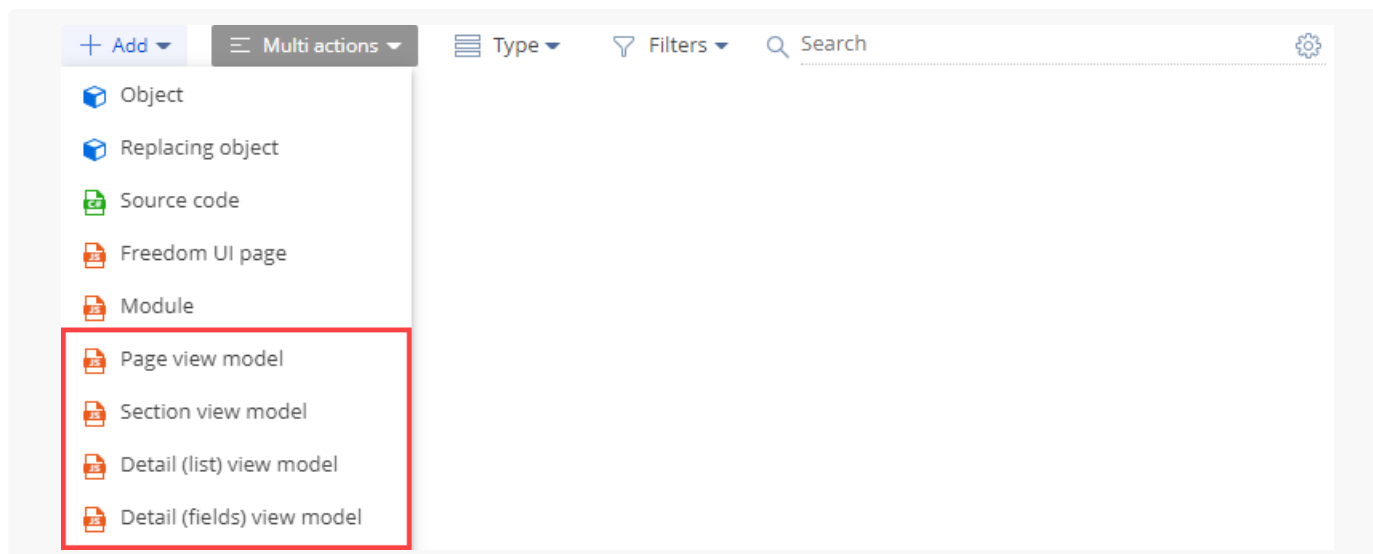
Для применения изменений свойств нажмите [ *Применить* ] ([ *Apply* ]).

#### **Дополнительные свойства** схемы:

- [ *Локализуемые строки* ] ([ *Localizable strings* ]).
  - [ *Изображения* ] ([ *Images* ]).
  - [ *Параметры* ] ([ *Parameters* ]).
6. В дизайнере модуля добавьте исходный код. Название модуля в функции `define()` должно совпадать с названием схемы (свойство [ *Код* ] ([ *Code* ])).
- Если при написании кода допущена ошибка, то слева возле номера строки отображается тип ошибки (ошибка  или предупреждение ). При наведении курсора на тип ошибки отображается всплывающая подсказка с текстовым описанием.
7. На панели инструментов дизайнера модуля нажмите [ *Сохранить* ] ([ *Save* ]).

## Схема модели представления

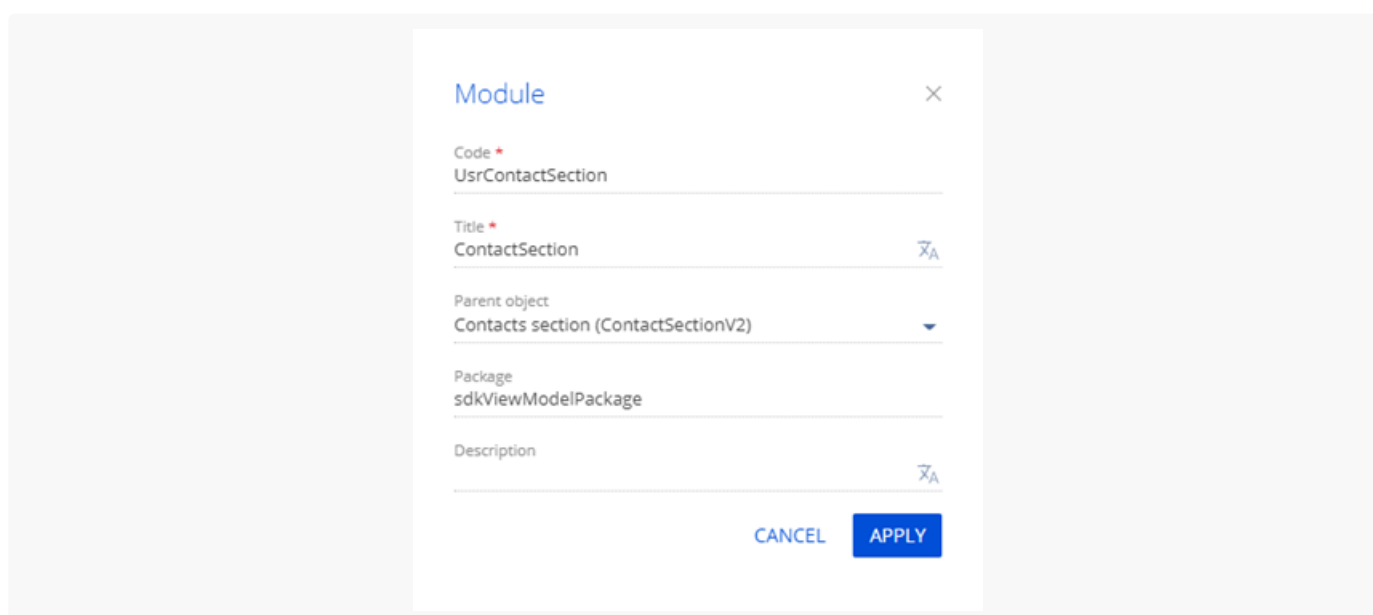
1. [Перейдите в раздел \[ \*Конфигурация\* \]](#) ([ *Configuration* ]) и выберите пользовательский [пакет](#), в который будет добавлена схема.
2. На панели инструментов реестра раздела нажмите [ *Добавить* ] ([ *Add* ]) и выберите вид схемы модели представления.



3. В дизайнере модуля заполните свойства схемы.



#### Основные свойства схемы:

- [ Код ] ([ Code ]) — название схемы (обязательное свойство). Начинается с префикса (по умолчанию `Usr`), который указан в системной настройке [ Префикс названия объекта ] ([ Prefix for object name ], код `SchemaNamePrefix`). Может содержать символы латинского алфавита и цифры.
- [ Заголовок ] ([ Title ]) — локализуемый заголовок схемы (обязательное свойство).
- [ Пакет ] ([ Package ]) — пользовательский пакет, в котором создается схема. Заполняется автоматически и недоступно для редактирования.
- [ Родительский объект ] ([ Parent object ]) — родительский объект для текущего модуля, свойства которого необходимо наследовать. В выпадающем списке выберите родительский объект, свойства которого необходимо наследовать.
- [ Описание ] ([ Description ]) — локализуемое описание схемы.



Для применения изменений свойств нажмите [ Применить ] ([ Apply ]).



**Панель свойств** дизайнера модуля позволяет:

- Изменить основные свойства схемы (кнопка .
- Задать дополнительные свойства схемы (кнопка .

**Дополнительные свойства** схемы:

- [ *Локализуемые строки* ] ([ *Localizable strings* ]).
- [ *Изображения* ] ([ *Images* ]).
- [ *Параметры* ] ([ *Parameters* ]).

4. В дизайнере модуля добавьте исходный код. Название модуля в функции `define()` должно совпадать с названием схемы (свойство [ *Код* ] ([ *Code* ])). Схема модели представления является наследником базовой схемы `BaseModulePageV2`.

Если при написании кода допущена ошибка, то слева возле номера строки отображается тип ошибки (ошибка  или предупреждение ). При наведении курсора на тип ошибки отображается всплывающая подсказка с текстовым описанием.

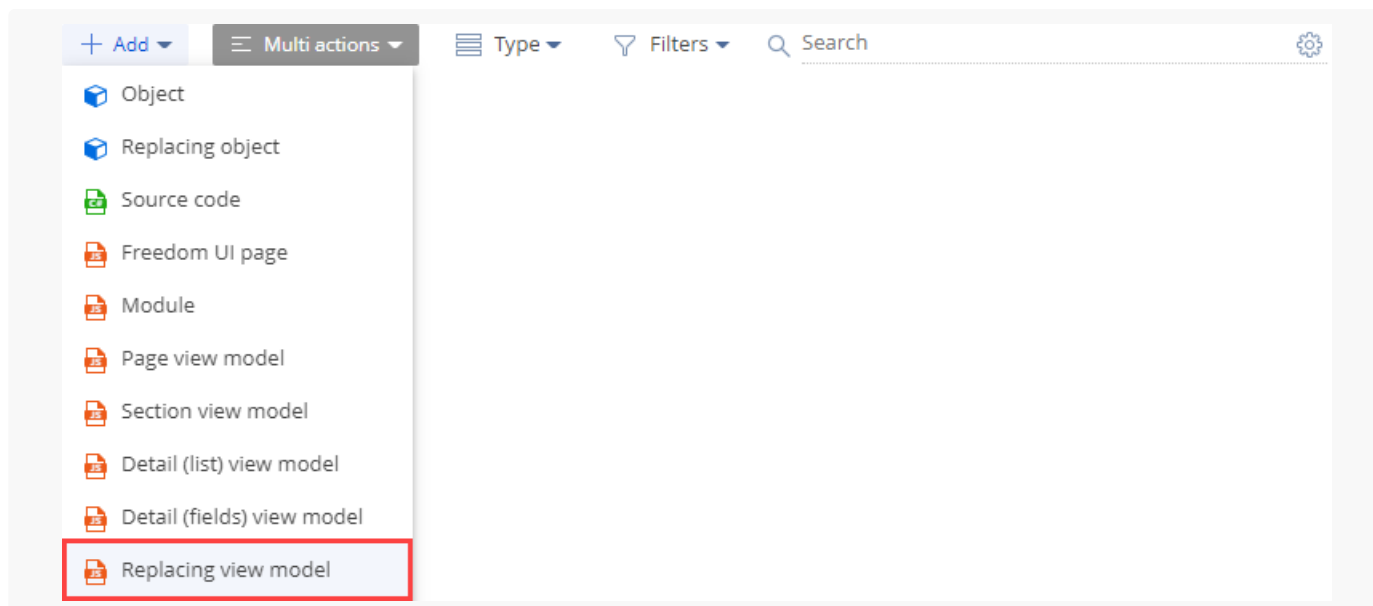
5. На панели инструментов дизайнера модуля нажмите [ *Сохранить* ] ([ *Save* ]).

## Реализовать замещающий модуль

Замещающий модуль представлен **видом** схемы [ *Замещающая модель представления* ] ([ *Replacing view model* ]).

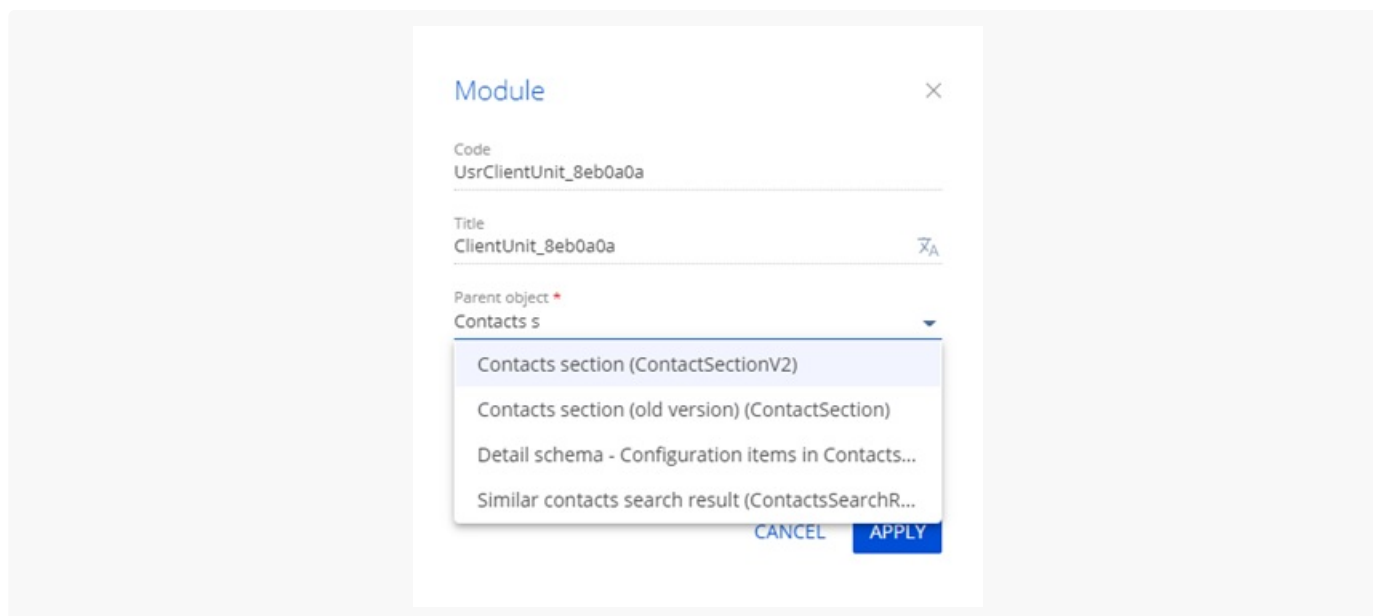
Чтобы **реализовать замещающий модуль**:

1. [Перейдите в раздел \[ \*Конфигурация\* \]](#) ([ *Configuration* ]) и выберите пользовательский [пакет](#), в который будет добавлена схема.
2. Установите [зависимости](#) пакета. В зависимости обязательно добавьте пакет, который содержит замещаемый клиентский модуль.
3. На панели инструментов реестра раздела нажмите [ *Добавить* ] —> [ *Замещающая модель представления* ] ([ *Add* ] —> [ *Replacing view model* ]).

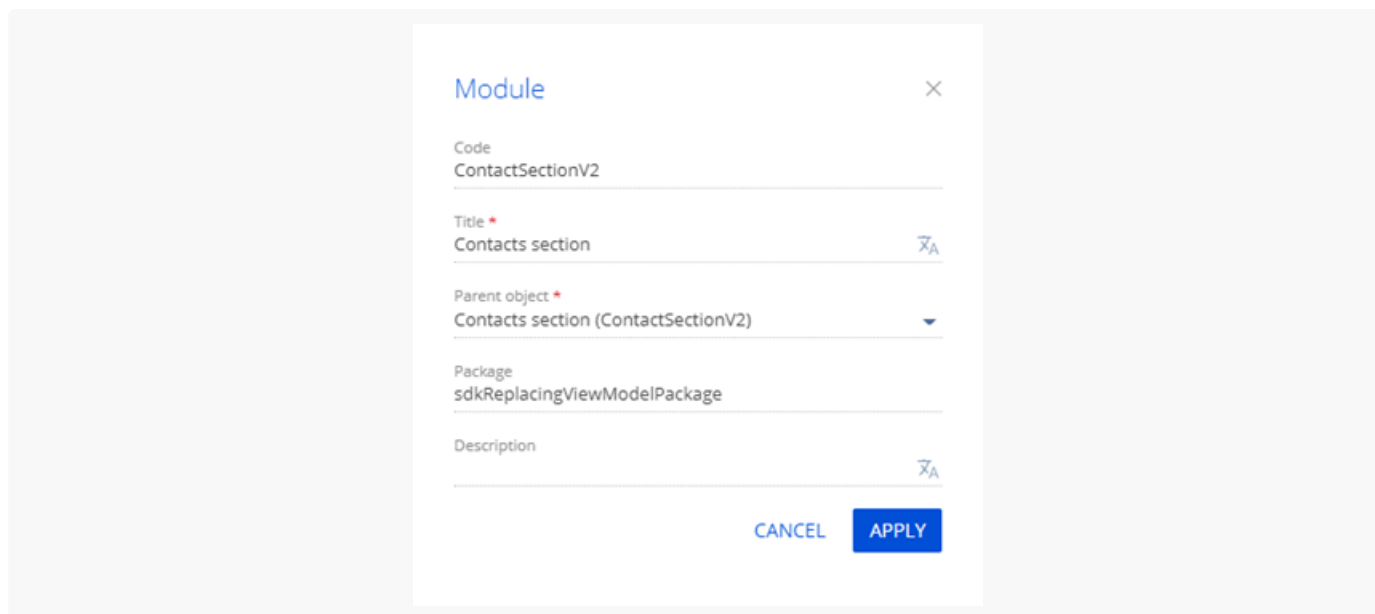


4. В дизайнере модуля выберите родительский объект.

Чтобы модуль замещал раздел или страницу, в выпадающем списке обязательного свойства [ Родительский объект ] ([ *Parent object* ]) схемы выберите схему модели представления, которую планируется заместить. Например, чтобы создать замещающую схему раздела [ Контакты ] ([ *Contacts* ]) в качестве родительского объекта укажите схему `ContactSectionV2`.



После выбора родительского объекта остальные свойства модуля заполняются автоматически.



Для применения заданных свойств нажмите [ Применить ] ([ Apply ]).

**Панель свойств** дизайнера модуля позволяет:

- Изменить основные свойства схемы (кнопка ).
- Задать дополнительные свойства схемы (кнопка ).

**Дополнительные свойства** схемы:

- [ Локализуемые строки ] ([ Localizable strings ]).
- [ Изображения ] ([ Images ]).
- [ Параметры ] ([ Parameters ]).

5. В дизайнере модуля добавьте исходный код. Исходный код должен реализовывать функциональность, которая отличает замещающий клиентский модуль от замещаемого. Название модуля в функции `define()` должно совпадать с названием схемы (свойство [ Код ] ([ Code ])).

Если при написании кода допущена ошибка, то слева возле номера строки отображается тип ошибки (ошибка или предупреждение ). При наведении курсора на тип ошибки отображается всплывающая подсказка с текстовым описанием.

6. На панели инструментов дизайнера модуля нажмите [ Сохранить ] ([ Save ]).

Подробнее о замещении конфигурационных элементов читайте в статье [Замещение конфигурационных элементов](#).

## Объект

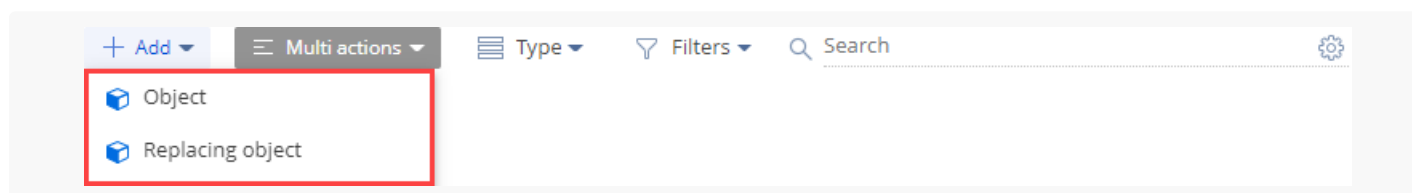
### Основы

**Конфигурационный элемент типа [ Объект ] ([ Object ])** — это бизнес-сущность, которая на уровне серверного ядра позволяет объявить новый класс ORM-модели. На уровне базы данных создание объекта означает создание записи таблицы с таким же именем, как у созданного объекта, и с таким же

набором колонок. То есть в большинстве случаев каждый объект в системе является системным представлением одной физической таблицы в базе данных. **Назначение** объекта — back-end разработка в приложении Creatio.

Виды объектов, которые позволяет добавить Creatio IDE, представлены значениями выпадающего списка [ *Добавить* ] ([ *Add* ]) панели инструментов рабочей области раздела [ *Конфигурация* ] ([ *Configuration* ]).

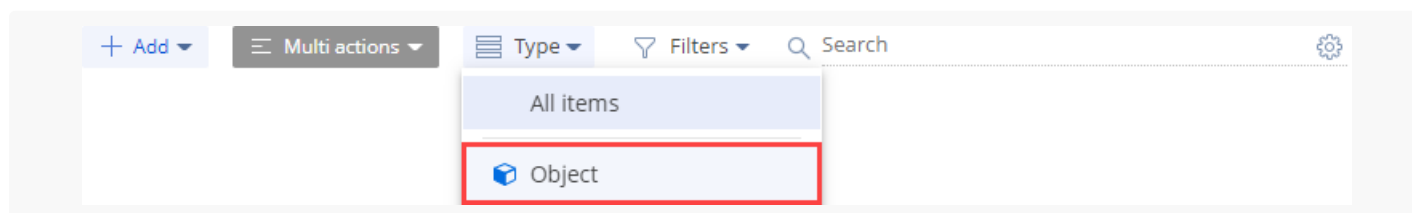
**Виды** объектов представлены на рисунке ниже.



Подробнее о видах конфигурационных элементов читайте в статье [Операции в Creatio IDE](#).

Объект представлен схемой типа [ *Объект* ] ([ *Object* ]) выпадающего списка [ *Тип* ] ([ *Type* ]) панели инструментов рабочей области раздела [ *Конфигурация* ] ([ *Configuration* ]). В схеме объекта описывается набор колонок, индексов и методов объекта. Платформа Creatio не ограничивает количество колонок объекта. Количество колонок в объекте ограничивается максимально допустимым количеством столбцов в таблицах базы данных, которую использует клиент.

**Тип** схемы объекта представлен на рисунке ниже.



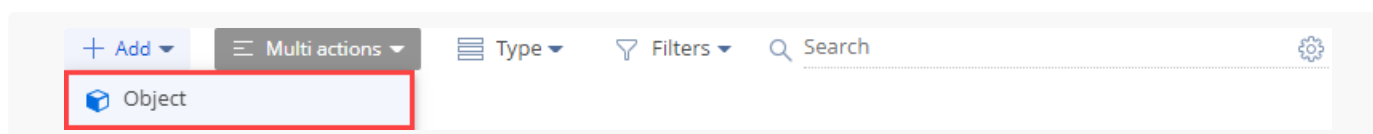
Подробнее о типах конфигурационных элементов читайте в статье [Операции в Creatio IDE](#).

## Реализовать объект

Объект представлен **видом** схемы [ *Объект* ] ([ *Object* ]).

Чтобы **реализовать объект**:

1. [Перейдите в раздел \[ \*Конфигурация\* \]](#) ([ *Configuration* ]) и выберите пользовательский [пакет](#), в который будет добавлена схема.
2. На панели инструментов реестра раздела нажмите [ *Добавить* ] —> [ *Объект* ] ([ *Add* ] —> [ *Object* ]).



3. В дизайнера объекта заполните свойства схемы.

**Основные свойства** схемы:

- [ *Код* ] ([ *Code* ]) — название схемы (обязательное свойство). Начинается с префикса (по умолчанию `Usr`), который указан в системной настройке [ *Префикс названия объекта* ] ([ *Prefix for object name* ], код `SchemaNamePrefix`). Может содержать символы латинского алфавита и цифры. Допустимая длина имени объекта — 128 символов. На базах Oracle ниже версии 12.2 не допускаются к использованию объекты с длиной имени более 30 символов.
- [ *Заголовок* ] ([ *Title* ]) — локализуемый заголовок схемы (обязательное свойство).
- [ *Пакет* ] ([ *Package* ]) — пользовательский пакет, в котором создается схема. Заполняется автоматически и недоступно для редактирования.
- [ *Описание* ] ([ *Description* ]) — локализуемое описание схемы.

#### 4. В дизайнере объекта выберите родительский объект.

Чтобы объект наследовал функциональность другого объекта, в выпадающем списке свойства [ *Родительский объект* ] ([ *Parent object* ]) схемы выберите схему объекта, функциональность которого планируется наследовать. Например, чтобы наследовать функциональность базовой схемы объекта `BaseEntity` в качестве родительского объекта укажите схему `BaseEntity`. В свойство [ *Унаследованные колонки* ] ([ *Inherited columns* ]) схемы объекта автоматически добавляются колонки, которые унаследованы от родительского объекта.

#### 5. В дизайнере объекта выберите колонку, которая соответствует идентификатору объекта.

[ *Идентификатор* ] ([ *Id* ]) — системная колонка, которая используется в качестве первичного ключа в таблице базы данных (обязательное свойство). Для этого воспользуйтесь инструкцией, которая приведена в пункте [Установить идентификатор объекта](#).



6. В дизайнере объекта добавьте индекс объекта (опционально). Для этого воспользуйтесь инструкцией, которая приведена в пункте [Добавить индекс объекта](#).
7. В дизайнере объекта настройте каскадную связь (опционально). Для этого воспользуйтесь инструкцией, которая приведена в пункте [Настроить каскадную связь](#).
8. На панели инструментов дизайнера объекта нажмите [ Опубликовать ] ([ Publish ]) для создания соответствующей таблицы в базе данных.

Кнопка [ Опубликовать ] ([ Publish ]) позволяет генерировать статический контент и обновлять структуру базы данных. При этом компиляция конфигурации не выполняется. Это позволяет ускорить разработку объектов и замещающих объектов. Компиляция при публикации объекта необходима, если при редактировании встроенного процесса объекта он был сохранен, но не был опубликован в дизайнере процессов. Чтобы компилировать конфигурацию, генерировать статический контент и обновлять структуру базы данных, в выпадающем меню кнопки [ Опубликовать ] ([ Publish ]) выберите [ Опубликовать и компилировать ] ([ Publish and compile ]).

## Установить идентификатор объекта

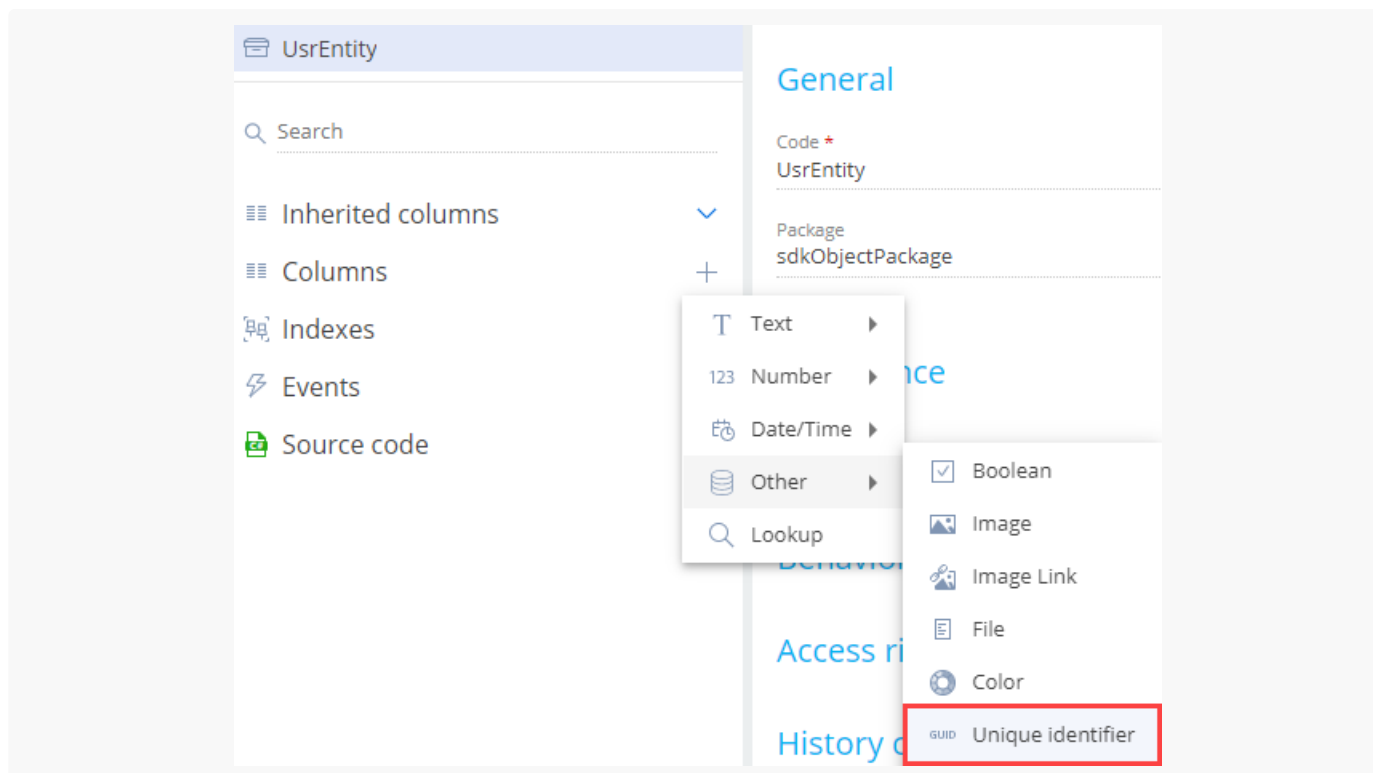
Поскольку объект в системе является представлением таблицы в базе данных, то он обязательно должен содержать колонку-идентификатор.

**Способы** установки идентификатора объекта:

- Если свойство [ Родительский объект ] ([ Parent object ]) — **базовый объект**, то свойство [ Идентификатор ] ([ Id ]) заполняется автоматически.
- Если свойство [ Родительский объект ] ([ Parent object ]) — **пользовательский объект**, то свойство [ Идентификатор ] ([ Id ]) заполняется вручную.

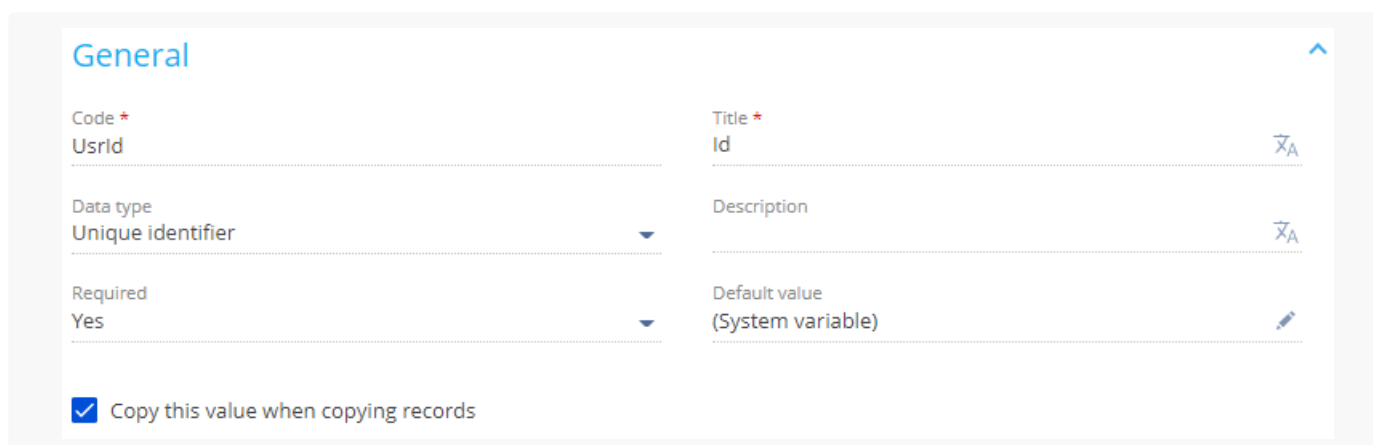
Чтобы **добавить пользовательский идентификатор**:

1. На панели свойств в контекстном меню узла [ Колонки ] ([ Columns ]) нажмите  $\oplus \rightarrow$  [ Другие ]  $\rightarrow$  [ Уникальный идентификатор ] ([ Other ]  $\rightarrow$  [ Unique identifier ]).



2. В дизайнере объекта заполните свойства колонки.

**Основные свойства** колонки:



- [ Код ] ([ Code ]) — название колонки (обязательное свойство). Начинается с префикса (по умолчанию `Usr`), который указан в системной настройке [ Префикс названия объекта ] ([ Prefix for object name ], код `SchemaNamePrefix`). Может содержать символы латинского алфавита и цифры. Значение свойства [ Код ] ([ Code ]) не должно совпадать со значением аналогичного поля родительского объекта колонки. В другом случае, при попытке опубликовать объект, отображается сообщение об ошибке.
- [ Заголовок ] ([ Title ]) — локализуемый заголовок колонки (обязательное свойство).
- [ Тип данных ] ([ Data type ]) — тип данных, которые содержатся в колонке. Заполняется автоматически в соответствии с типом колонки, которая была выбрана при добавлении колонки. Недоступно для редактирования.
- [ Описание ] ([ Description ]) — локализуемое описание колонки.


- [ *Обязательное* ] ([ *Required* ]) — обязательность колонки. Выберите "да" ("Yes"), поскольку колонка типа [ *Уникальный идентификатор* ] ([ *Unique identifier* ]) является обязательной для объекта. При попытке сохранить схему объекта без колонки типа [ *Уникальный идентификатор* ] ([ *Unique identifier* ]) отображается соответствующее сообщение.
- [ *Значение по умолчанию* ] ([ *Default value* ]) — значение колонки по умолчанию. Чтобы **установить значение по умолчанию**, воспользуйтесь инструкцией, которая приведена в пункте [Установить значение по умолчанию для колонки типа \[ \*Уникальный идентификатор\* \]](#).
- [ *Режим использования* ] ([ *Usage mode* ]) — выберите "Расширенный" ("Advanced").

**Режимы использования** колонок, которые доступны в Creatio IDE:

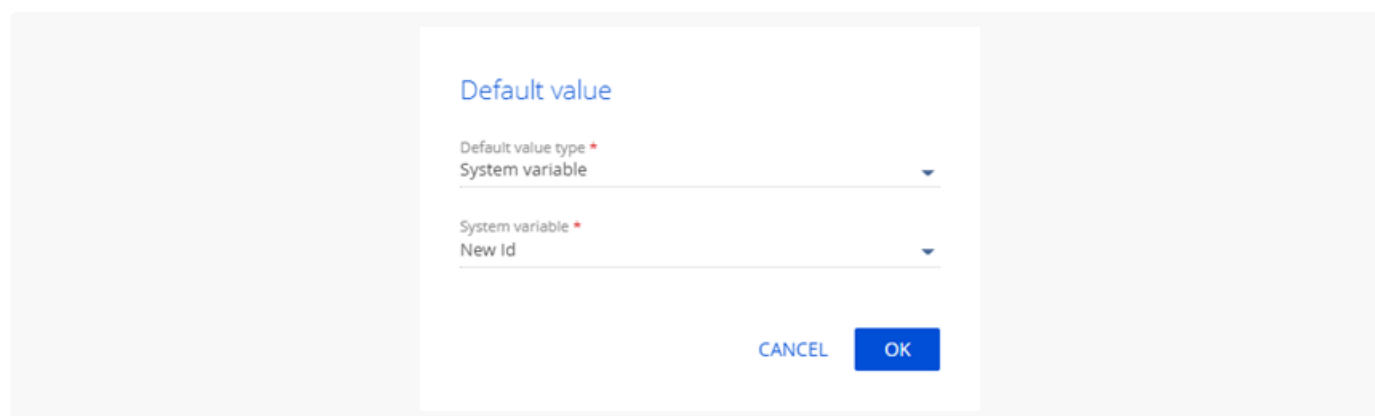
- [ *Общие* ] ([ *General* ]) — стандартный режим колонок в приложении.
- [ *Расширенный* ] ([ *Advanced* ]) — колонка отображается в конфигурации и доступна для использования в приложении.
- [ *Никогда* ] ([ *None* ]) — колонка отображается в конфигурации как системная и недоступна для использования в приложении.

## Установить значение по умолчанию для колонки типа [ *Уникальный идентификатор* ]

Чтобы **установить значение по умолчанию** для колонки типа [ *Уникальный идентификатор* ] ([ *Unique identifier* ]):

1. В свойстве [ *Значение по умолчанию* ] ([ *Default value* ]) нажмите на .
2. Заполните **свойства значения по умолчанию**:

- [ *Тип значения* ] ([ *Default value type* ]) — выберите "Системная переменная" ("System variable").
- [ *Системная переменная* ] ([ *System variable* ]) — выберите "Новый идентификатор" ("New Id"), поскольку идентификаторы должны быть уникальными.

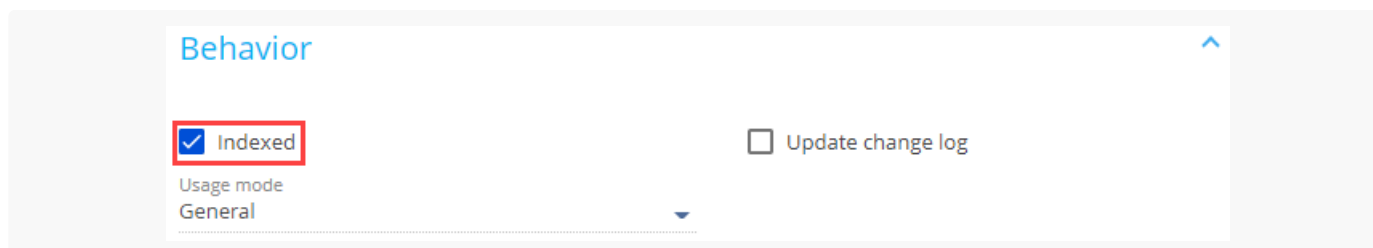


## Добавить индекс объекта

Кроме колонок Creatio IDE позволяет добавить в объект индексы, которые при публикации объекта будут автоматически созданы в таблице базы данных.

**Способы** добавления индекса объекта:

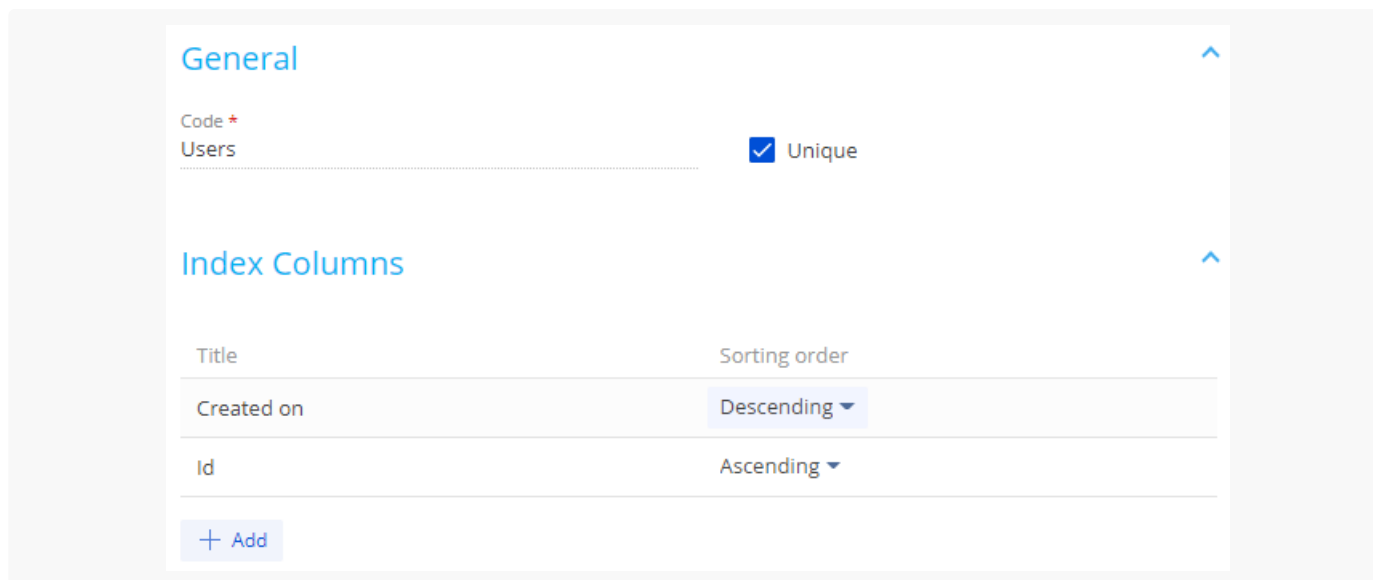
- **Индекс по одной колонке.** В этом случае в блоке свойств [ *Поведение* ] ([ *Behavior* ]) установите признак [ *Индексируемая* ] ([ *Indexed* ]). По умолчанию справочные колонки — индексируемые.



- **Составной индекс.** Для этого воспользуйтесь инструкцией, которая приведена ниже.

Чтобы **добавить составной индекс объекта**:

1. На панели свойств в контекстном меню узла [ *Индексы* ] ([ *Indexes* ]) нажмите **+**.
2. Заполните **свойства индекса**:
  - [ *Код* ] ([ *Code* ]) — название индекса (обязательное свойство).
  - [ *Уникальный* ] ([ *Unique* ]) — установите признак, если для колонок индекса необходимо задать ограничение целостности (исключить возможность вставки повторяющихся комбинаций значений).
  - [ *Колонки индекса* ] ([ *Index Columns* ]) — выберите колонки, которые необходимо добавить в индекс. Для этого в блоке свойств [ *Колонки индекса* ] ([ *Index Columns* ]) нажмите [ *Добавить* ] ([ *Add* ]), выберите колонку объекта и укажите направление сортировки.

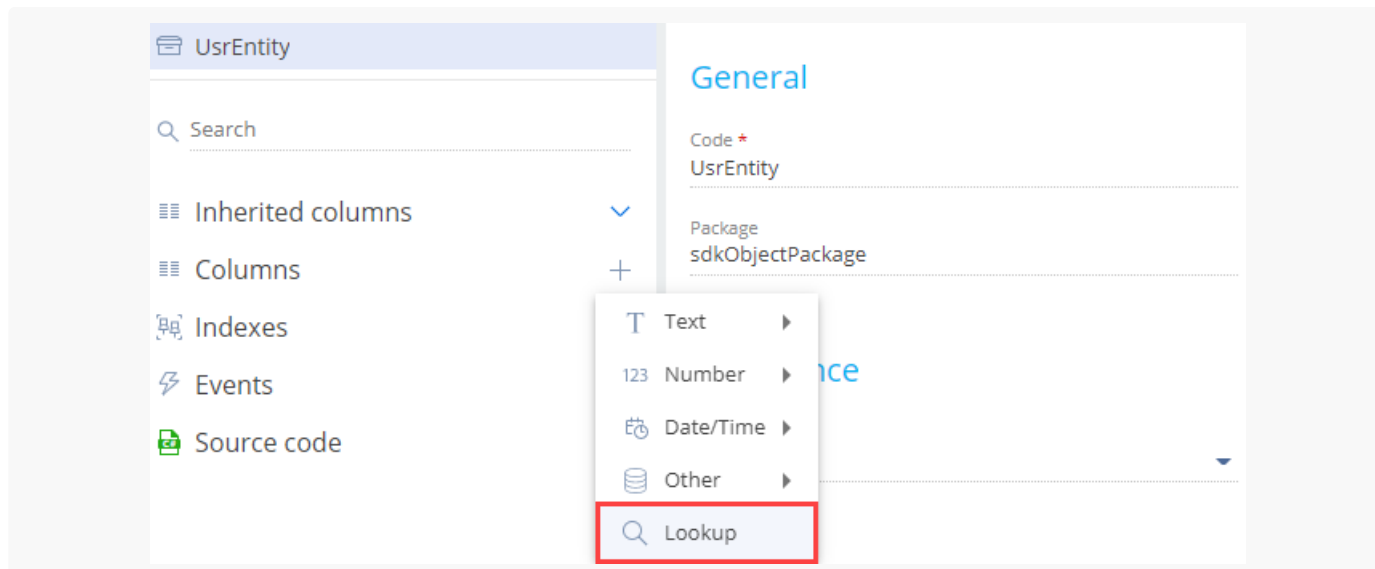


## Настроить каскадную связь

Creatio IDE позволяет настроить каскадную связь только для колонки типа [ *Справочник* ] ([ *Lookup* ]).

Чтобы **настроить каскадную связь**:

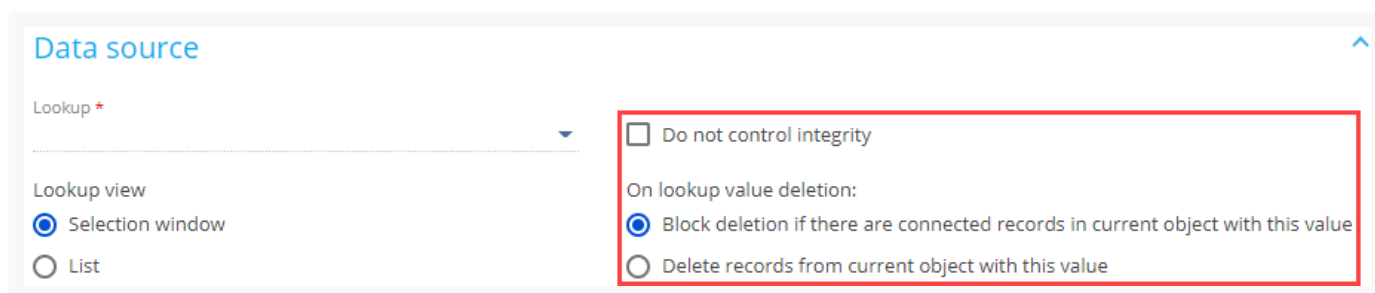
1. Добавьте колонку типа [ *Справочник* ] ([ *Lookup* ]) (опционально). Для этого на панели свойств в контекстном меню узла [ *Колонки* ] ([ *Columns* ]) нажмите **+** —> [ *Справочник* ] ([ *Lookup* ]).



2. В блоке свойств [ *Источник данных* ] ([ *Data source* ]) настройте **каскадную связь**.

**Способы** настройки каскадной связи:

- Признак [ *Не контролировать целостность* ] ([ *Do not control integrity* ]).
- Опции пункта [ *При удалении значения справочника* ] ([ *On lookup value deletion* ]).



Рассмотрим работу каскадной связи на примере объекта [ *Контакт* ] ([ *Contact* ]), который по справочной колонке [ *AccountId* ] связан с объектом [ *Контрагент* ] ([ *Account* ]). Для этого в поле [ *Выбор объекта* ] ([ *Lookup* ]) выберите [ *Account* ].

**Варианты настройки** каскадной связи:

- Если **установлен признак** [ *Не контролировать целостность* ] ([ *Do not control integrity* ]), то выполняется удаление контрагента. При этом не удаляются контакты, которые связаны с текущим контрагентом.
- Если **не установлен признак** [ *Не контролировать целостность* ] ([ *Do not control integrity* ]) и **выбрана опция** [ *Блокировать удаление, если есть связанные записи в текущем объекте с этим значением* ] ([ *Block deletion if there are connected records in current object with this value* ]), то удаление контрагента заблокировано, если присутствуют контакты, которые связаны с текущим контрагентом. В этом случае приложение выдает предупреждающее сообщение. После подтверждения выполняется удаление контрагента. При этом не удаляются контакты, которые связаны с текущим контрагентом.
- Если **не установлен признак** [ *Не контролировать целостность* ] ([ *Do not control integrity* ]) и **выбрана опция** [ *Удалять записи из текущего объекта с этим значением* ] ([ *Delete records from current object with this value* ]), то выполняется удаление контрагента вместе с удалением контактов,

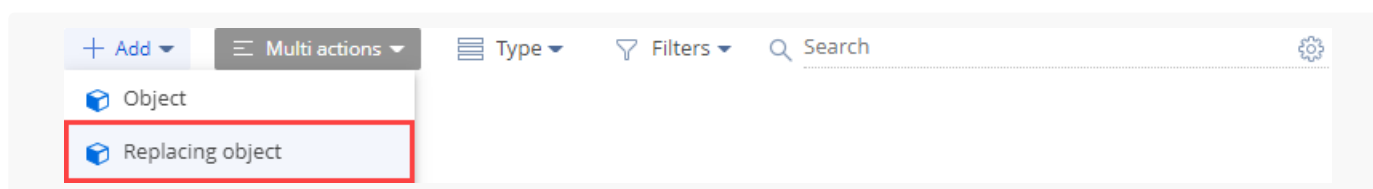
которые связаны с текущим контрагентом.

## Реализовать замещающий объект

Замещающий объект представлен **видом** схемы [ *Замещающий объект* ] ([ *Replacing object* ]).

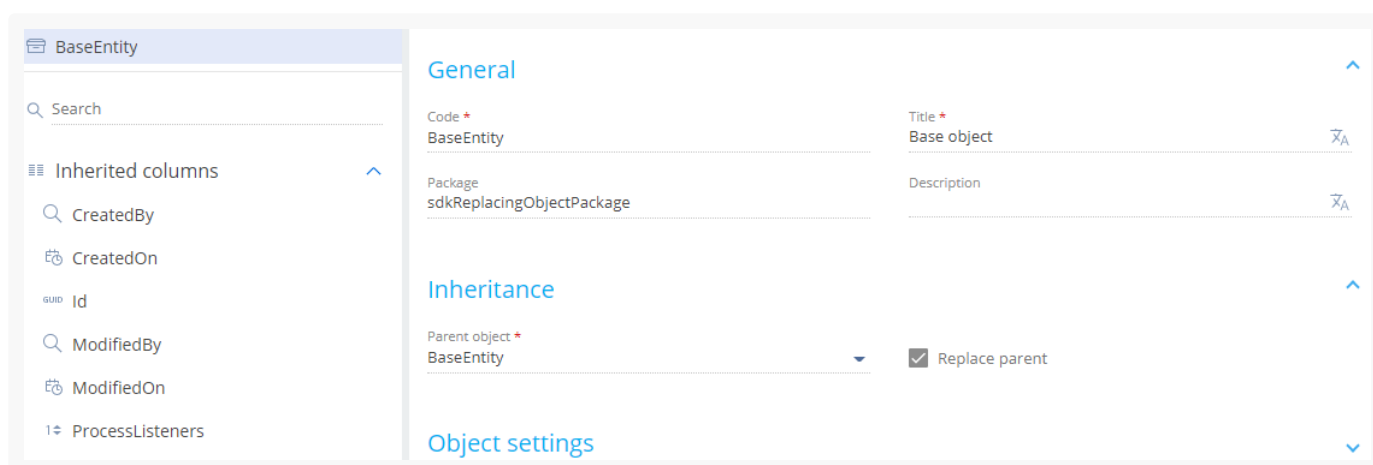
Чтобы **реализовать замещающий объект**:

1. [Перейдите в раздел \[ Конфигурация \]](#) ([ *Configuration* ]) и выберите пользовательский [пакет](#), в который будет добавлена схема.
2. Установите [зависимости](#) пакета. В зависимости обязательно добавьте пакет, который содержит замещаемый объект.
3. На панели инструментов реестра раздела нажмите [ *Добавить* ] —> [ *Замещающий объект* ] ([ *Add* ] —> [ *Replacing object* ]).



4. В дизайнера объекта выберите родительский объект.

Чтобы объект замещал функциональность базового объекта, в выпадающем списке обязательного свойства [ *Родительский объект* ] ([ *Parent object* ]) схемы выберите схему объекта, которую планируется заместить. Например, чтобы замещать функциональность базовой схемы объекта `BaseEntity` в качестве родительского объекта укажите схему `BaseEntity`. В свойство [ *Унаследованные колонки* ] ([ *Inherited columns* ]) схемы объекта автоматически добавляются колонки, которые унаследованы от родительского объекта. После выбора родительского объекта остальные свойства объекта заполняются автоматически.



5. В дизайнера объекта реализуйте функциональность, которая отличает замещающий объект от замещаемого.
6. На панели инструментов дизайнера объекта нажмите [ *Опубликовать* ] ([ *Publish* ]) для создания соответствующей таблицы в базе данных.

Кнопка [ *Опубликовать* ] ([ *Publish* ]) позволяет генерировать статический контент и обновлять

структуру базы данных. При этом компиляция конфигурации не выполняется. Это позволяет ускорить разработку объектов и замещающих объектов. Компиляция при публикации объекта необходима, если при редактировании встроенного процесса объекта он был сохранен, но не был опубликован в дизайнера процессов. Чтобы компилировать конфигурацию, генерировать статический контент и обновлять структуру базы данных, в выпадающем меню кнопки [ *Опубликовать* ] ([ *Publish* ]) выберите [ *Опубликовать и компилировать* ] ([ *Publish and compile* ]).

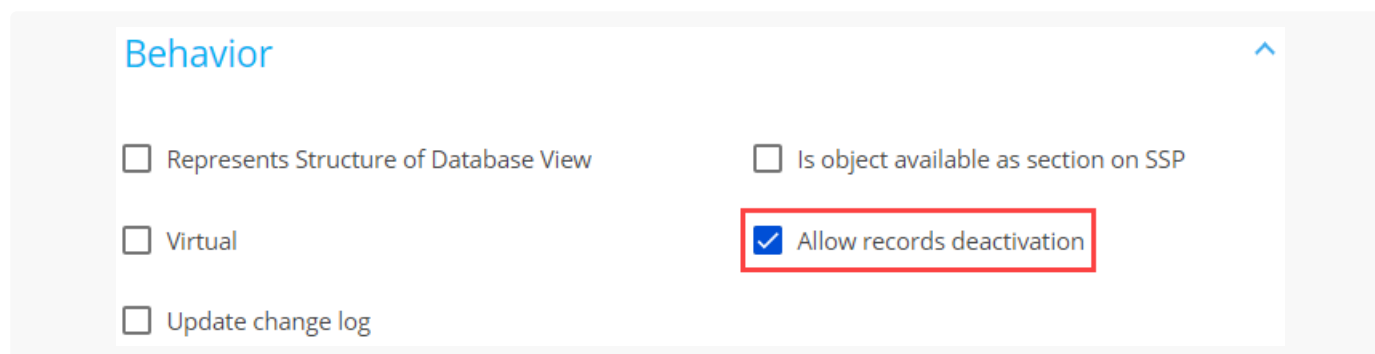
Подробнее о замещении конфигурационных элементов читайте в статье [Замещение конфигурационных элементов](#).

## Деактивировать записи объекта

Creatio предоставляет возможность деактивировать записи объектов приложения для исключения их из бизнес-логики. Это может понадобиться, например, если данные устарели и больше не используются. Функциональность деактивации записей объектов доступна для всех объектов.

Чтобы **деактивировать записи объекта**:

1. Откройте схему объекта, записи которой планируется деактивировать.
2. В блоке свойств [ *Поведение* ] ([ *Behavior* ]) установите признак [ *Разрешить деактивацию записей* ] ([ *Allow records deactivation* ]).



3. На панели инструментов дизайнера объекта нажмите [ *Опубликовать* ] ([ *Publish* ]).

Автоматическая **фильтрация неактивных записей** доступна для отдельных элементов интерфейса.

Элементы интерфейса, для которых **доступна автоматическая фильтрация записей**:

- Выпадающий список.
- Окно выбора значения из справочника.
- Быстрый фильтр.

Элементы интерфейса, для которых **недоступна автоматическая фильтрация записей**:

- Страница с содержимым справочников.
- Раздел.
- Расширенный фильтр.

Параметр `UseRecordDeactivation` класса `EntitySchemaQuery` позволяет управлять фильтрацией по неактивным записям. По умолчанию — `false`. Если для параметра `UseRecordDeactivation` установить

значение `true`, то в запрос на выборку данных из объекта, где включена деактивация записей, добавляется фильтр, который исключает неактивные записи.

Пример деактивации записей на front-end стороне приведен ниже.

#### Пример деактивации записей (front-end)

```
var esq = Ext.create("Terrasoft.EntitySchemaQuery", {
    rootSchemaName: "MyCustomLookup",
    useRecordDeactivation: true
});
```

Пример деактивации записей на back-end стороне приведен ниже.

#### Пример деактивации записей (back-end)

```
var esq = new EntitySchemaQuery(userConnection.EntitySchemaManager, "ContactType") {
    UseRecordDeactivation = true
};
esq.PrimaryQueryColumn.IsAlwaysSelect = true;
```

Результирующий SQL-запрос приведен ниже.

#### Пример SQL-запроса

```
SELECT [ContactType].[Id] [Id]
FROM [dbo].[ContactType] [ContactType] WITH(NOLOCK)
WHERE [ContactType].[RecordInactive] = 0
```

## Исходный код (C#)

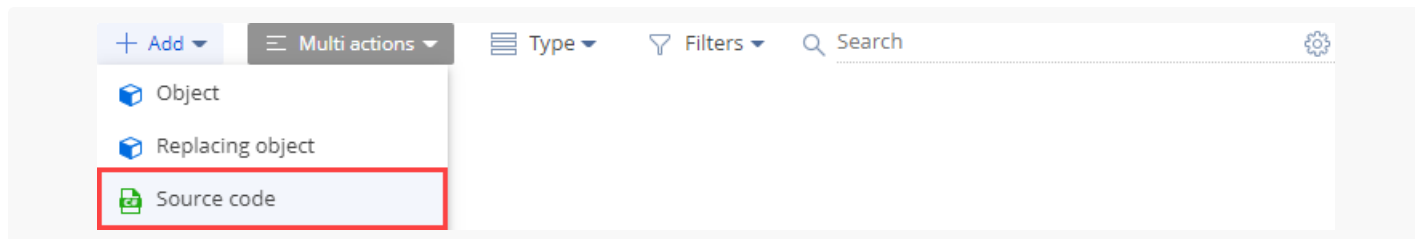


Основы

**Конфигурационный элемент типа** [ *Исходный код* ] ([ *Source code* ]) — это сущность, которая реализует бизнес-логику. Позволяет добавлять, удалять и форматировать исходный код добавляемой функциональности на языке C#. **Назначение** конфигурационного элемента типа [ *Исходный код* ] ([ *Source code* ]) — back-end разработка в приложении Creatio.

Схема исходного кода, которую позволяет добавить Creatio IDE, представлена значением выпадающего списка [ *Добавить* ] ([ *Add* ]) панели инструментов рабочей области раздела [ *Конфигурация* ] ([ *Configuration* ]).

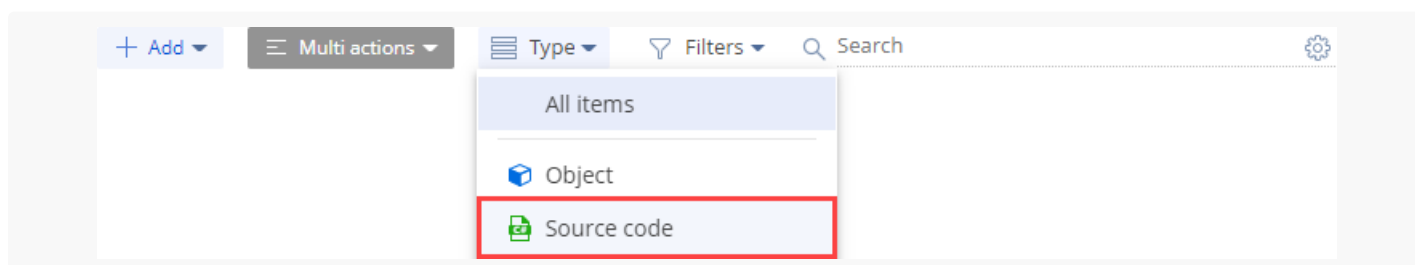




Подробнее о видах конфигурационных элементов читайте в статье [Операции в Creatio IDE](#).

Конфигурационный элемент типа [ Исходный код ] ([ Source code ]) представлен схемой типа [ Исходный код ] ([ Source code ]) выпадающего списка [ Тип ] ([ Type ]) панели инструментов рабочей области раздела [ Конфигурация ] ([ Configuration ]). **Схема** — основа конфигурации Creatio.

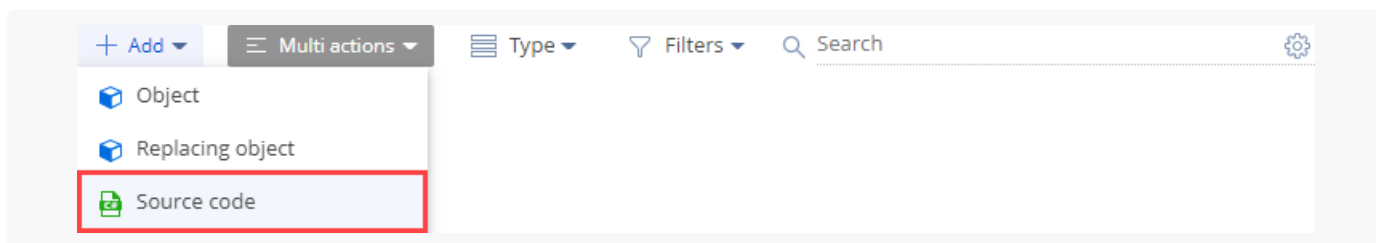
**Тип** схемы исходного кода представлен на рисунке ниже.



Подробнее о типах конфигурационных элементов читайте в статье [Операции в Creatio IDE](#).

## Реализовать исходный код

1. [Перейдите в раздел \[ Конфигурация \]](#) ([ Configuration ]) и выберите пользовательский [пакет](#), в который будет добавлена схема.
2. На панели инструментов реестра раздела нажмите [ Добавить ] —> [ Исходный код ] ([ Add ] —> [ Source code ]).

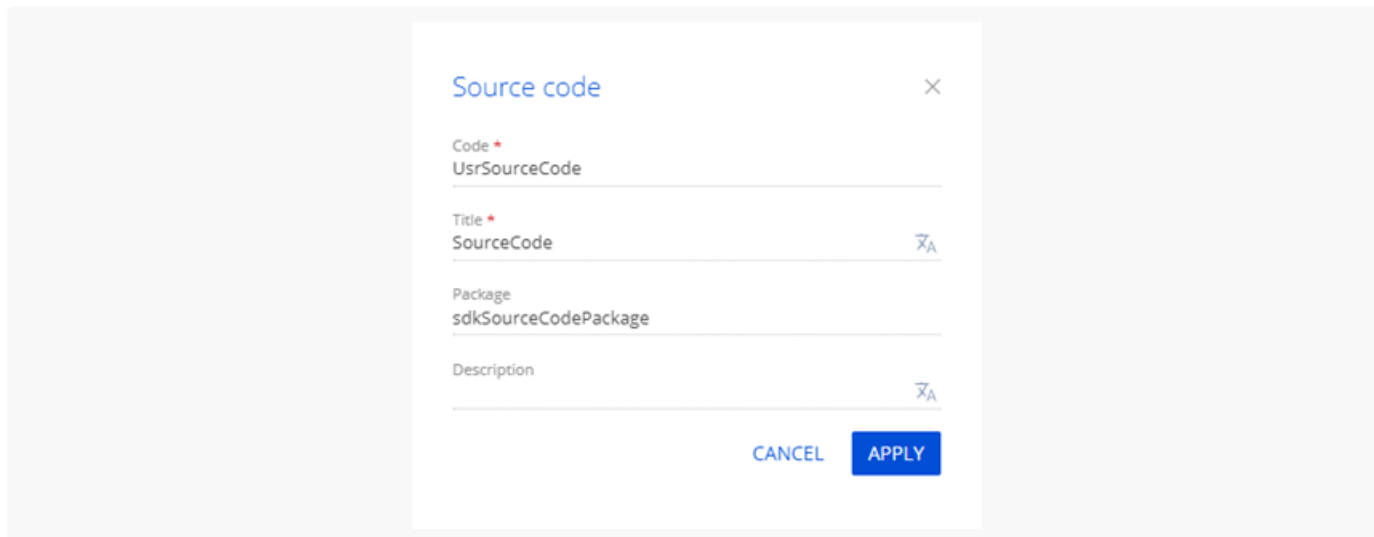


3. В дизайнера исходного кода заполните свойства схемы.

### Основные свойства схемы:

- [ Код ] ([ Code ]) — название схемы (обязательное свойство). Начинается с префикса (по умолчанию `usr`), который указан в системной настройке [ Префикс названия объекта ] ([ Prefix for object name ], код `SchemaNamePrefix`). Может содержать символы латинского алфавита и цифры.
- [ Заголовок ] ([ Title ]) — локализуемый заголовок колонки (обязательное свойство).
- [ Пакет ] ([ Package ]) — пользовательский пакет, в котором создается схема. Заполняется автоматически и недоступно для редактирования.

- [ Описание ] ([ Description ]) — локализуемое описание колонки.



Для применения изменений свойств нажмите [ Применить ] ([ Apply ]).

**Панель свойств** дизайнера исходного кода позволяет:

- Изменить основные свойства схемы (кнопка ).
- Задать дополнительные свойства схемы (кнопка ).

**Дополнительные свойства** схемы: [ Локализуемые строки ] ([ Localizable strings ]).

4. В дизайнера исходного кода добавьте исходный код. Название класса, который объявлен в исходном коде, должно совпадать с названием схемы (свойство [ Код ] ([ Code ])).

Если при написании кода допущена ошибка, то слева возле номера строки отображается тип ошибки (ошибка или предупреждение ). При наведении курсора на тип ошибки отображается всплывающая подсказка с текстовым описанием.

В дизайнера исходного кода Creatio IDE позволяет создать замещающий класс. Для этого воспользуйтесь инструкцией, которая приведена в пункте [Реализовать замещающий класс](#).

5. На панели инструментов дизайнера исходного кода нажмите [ Сохранить ] ([ Save ]) для временного сохранения изменений в метаданных схемы.
6. На панели инструментов дизайнера исходного кода нажмите [ Опубликовать ] ([ Publish ]) для выполнения изменений на уровне базы данных.

## Реализовать замещающий класс

Принцип замещения классов, в частности создание и использование в конфигурации экземпляров замещаемых классов, имеет свои особенности.

Чтобы **реализовать замещающий класс**:

1. Создайте класс, который является наследником замещаемого класса.
2. Для класса добавьте атрибут `[override]`. Описание атрибута содержится в статье [Атрибут \[override\]](#).
3. Реализуйте функциональность, которая отличает замещающий класс от замещаемого. Например,

реализуйте свойства и методы, которые расширяют функциональность замещаемого класса, перегрузку методов замещаемого класса и т. д.).

- В замещающем классе для свойств и методов добавьте модификатор `override`.
- В пользовательском замещаемом классе для свойств и методов, которые необходимо заместить, добавьте модификатор `virtual`.

В базовом классе можно заместить только виртуальные методы или реализовать абстрактные методы. До компиляции все замещающие свойства и методы, которые объявлены без использования ключевого слова `override`, недоступны. Привязка и внедрение зависимостей типов выполняется open-source фреймворком внедрения зависимостей [Ninject](#) только во время выполнения.

Подробнее о замещении конфигурационных элементов читайте в статье [Замещение конфигурационных элементов](#).

# Замещение конфигурационных элементов

## Основы

Разработка на платформе Creatio базируется на основных принципах объектно-ориентированного программирования. В частности, модель расширения Creatio основана на **принципе открытости-закрытости**, при котором основная логика приложения закрыта для изменения напрямую, но открыта для расширения. Это означает, что функциональность необходимо разрабатывать путем добавления новых сущностей, а не путем изменения существующих.

Конфигурационные элементы, которые расположены в предустановленных пакетах, недоступны для изменения на уровне системы. Разработку и модификацию функциональности необходимо выполнять в пользовательских [пакетах](#) с использованием **механизма замещения**. При реализации замещения в Creatio используются понятия замещающего и замещаемого конфигурационных элементов.

**Замещающий конфигурационный элемент** — конфигурационный элемент, который замещает другой конфигурационный элемент соответствующего типа.

**Замещаемый конфигурационный элемент** — конфигурационный элемент, который замещен другим конфигурационным элементом соответствующего типа.

**Конфигурационные элементы**, которые позволяет заместить Creatio IDE:

- **Клиентский модуль, который определяет модель представления.**

Клиентский модуль реализует front-end часть приложения. Чтобы создать замещающий клиентский модуль, необходимо использовать схему замещающей модели представления. Для этого воспользуйтесь инструкцией, которая приведена в статье [Клиентский модуль](#).

- **Объект.**

Объект реализует back-end часть приложения. Чтобы создать замещающий объект, необходимо использовать схему замещающего объекта. Для этого воспользуйтесь инструкцией, которая приведена в статье [Объект](#).

- **Исходный код.**

Исходный код реализует back-end часть приложения. В качестве замещающего конфигурационного

элемента выступает класс. Чтобы создать замещающий класс, необходимо использовать схему типа [ *Исходный код* ] ([ *Source code* ]). Для этого воспользуйтесь инструкцией, которая приведена в статье [Исходный код \(C#\)](#).

После реализации замещающего конфигурационного элемента при обращении к нему приложение выполняет логику замещающего конфигурационного элемента.

Creatio IDE позволяет выполнять замещение одного и того же конфигурационного элемента в нескольких пользовательских пакетах. При этом конечная реализация замещающего конфигурационного элемента в скомпилированной конфигурации определяется [иерархией пакетов](#), которые содержат замещающие конфигурационные элементы.