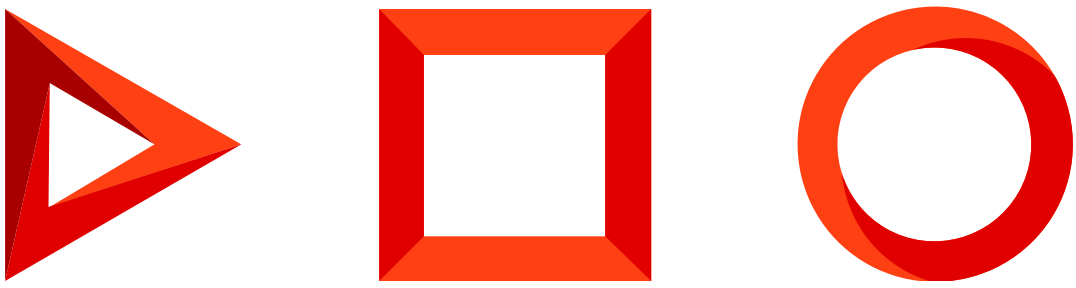
Creatio Academy

Элементы интерфейса

Модальное окно

Версия 7.18



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

Модальное окно

Основы

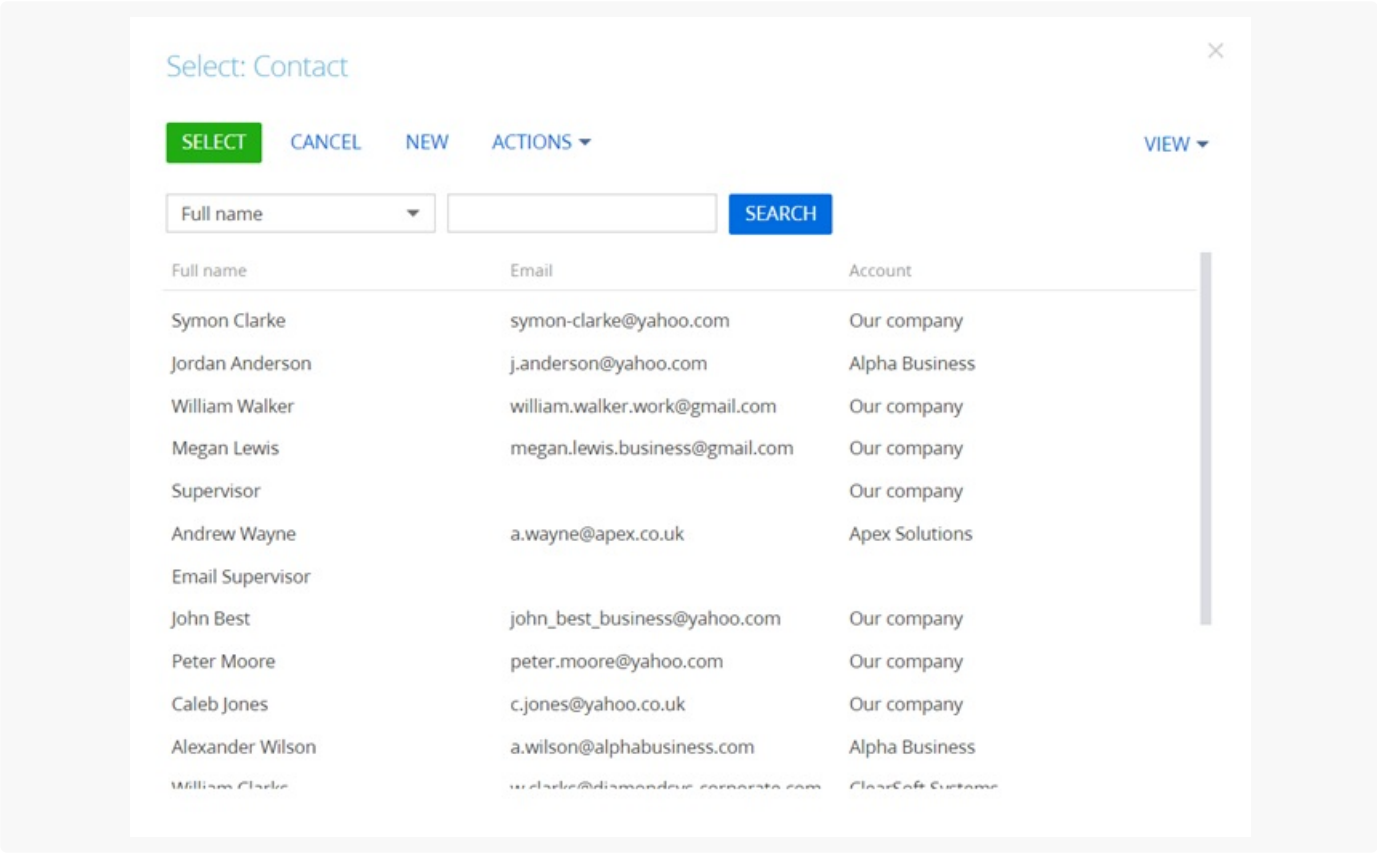
Назначение модального окна — отображение данных во всплывающем диалоговом окне.

Логика работы модального окна:

- Остается открытой страница приложения, из которой было вызвано модальное окно.
- Не выполняется переход на новую страницу приложения.
- Страница, которая отображается в модальном окне, не учитывается в истории переходов по страницам браузера.

Модальное окно **позволяет**:

- Отображать произвольную информацию (например, текст, кнопки и т. д.).
- Выбирать данные из справочника. Например, выбор ответственного за активность выполняется из справочника контактов, который отображается в модальном окне.



Составляющие модального окна:

- Функциональность модального окна — схемы `ModalBox` и `ModalBoxSchemaModule` пакета `NUI`.
- Вызов модального окна для выбора данных из справочника — схема `LookupUtilitiesV2` пакета `NUI`.

Реализовать модальное окно

Сложный

Пример. Реализовать действие процесса для отображения модального окна в бизнес-процессе. Модальное окно содержит произвольный текст и кнопки [*Yes*], [*No*], [*Cancel*].

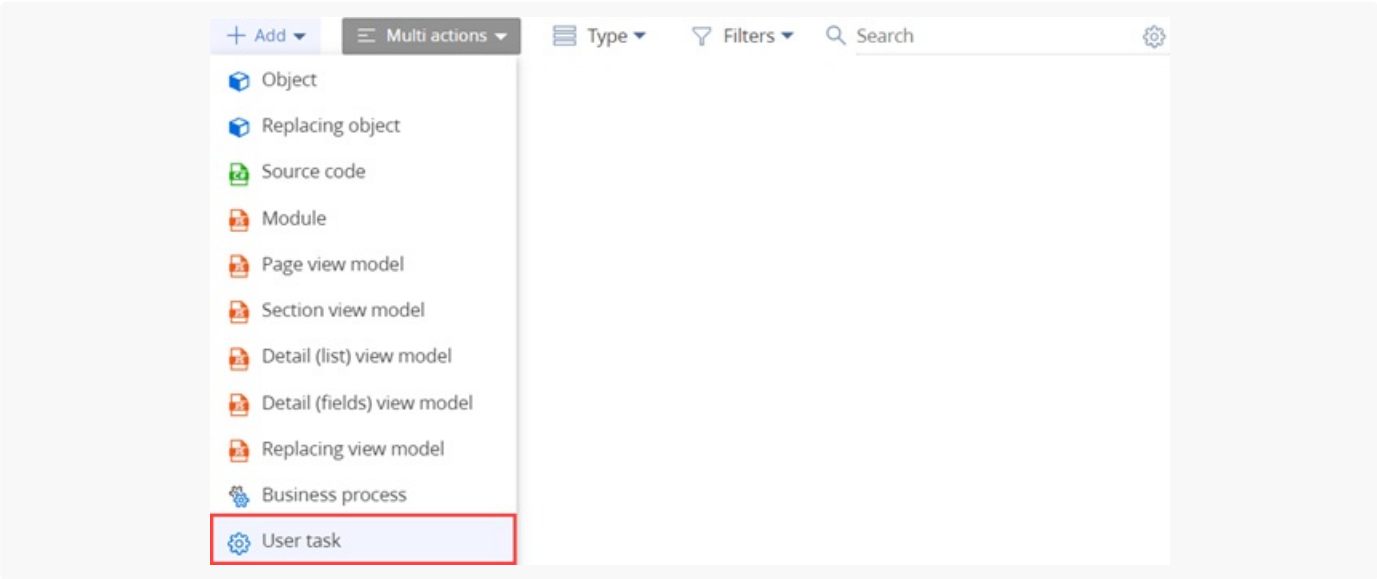
Логика работы кнопок:

- По нажатию на кнопки [*Yes*] и [*No*] отображаются соответствующие автогенерируемые страницы.
- По нажатию на кнопку [*Cancel*] закрывается модальное окно.

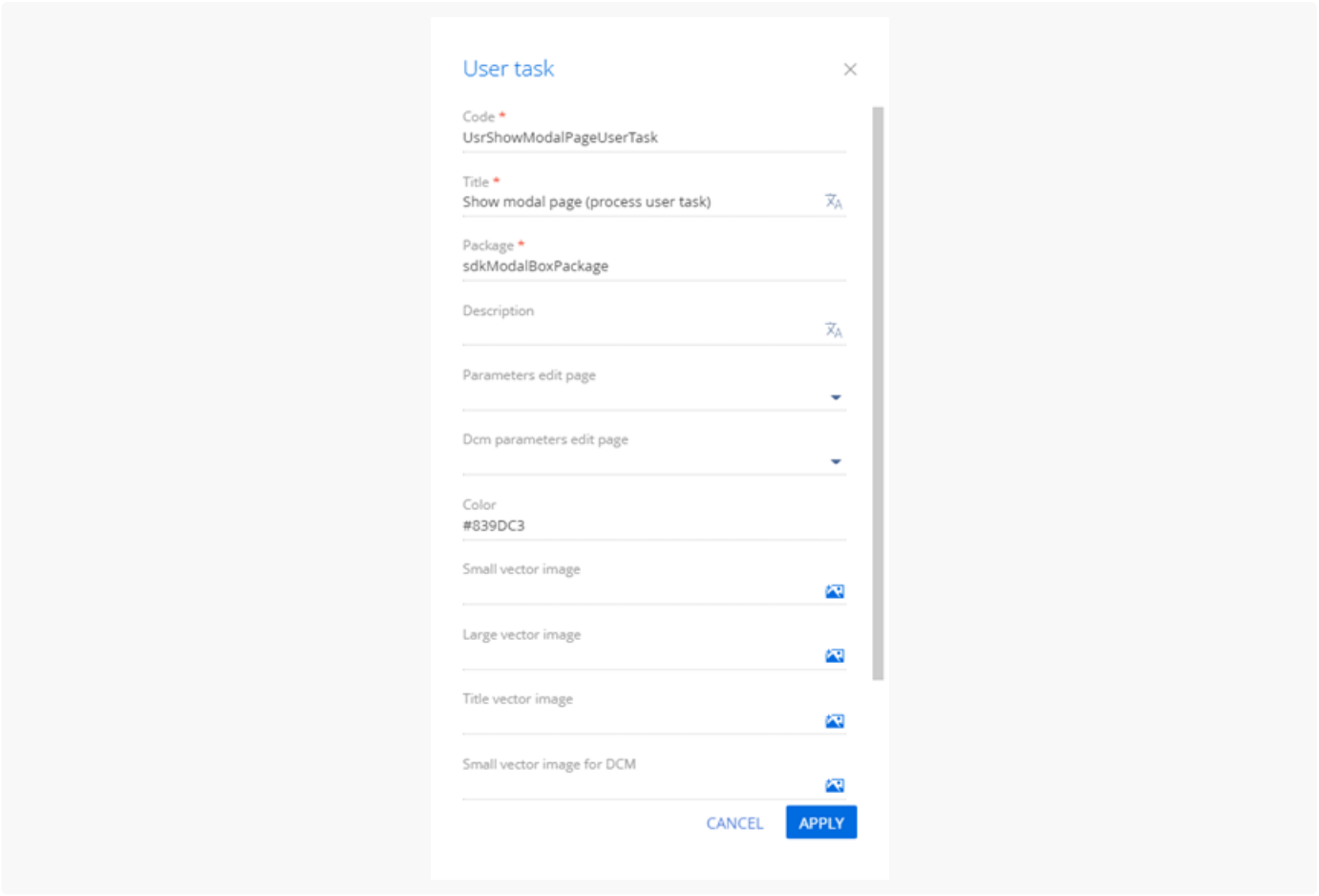
Для действия процесса настройте логирование.

1. Создать действие 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. Заполните **свойства действия процесса**.
   - [*Код*] ([*Code*]) — "`UsrShowModalPageUserTask`".
   - [*Заголовок*] ([*Title*]) — "`Показать модальное окно (действие процесса)`"("`Show modal page (process user task)`").



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

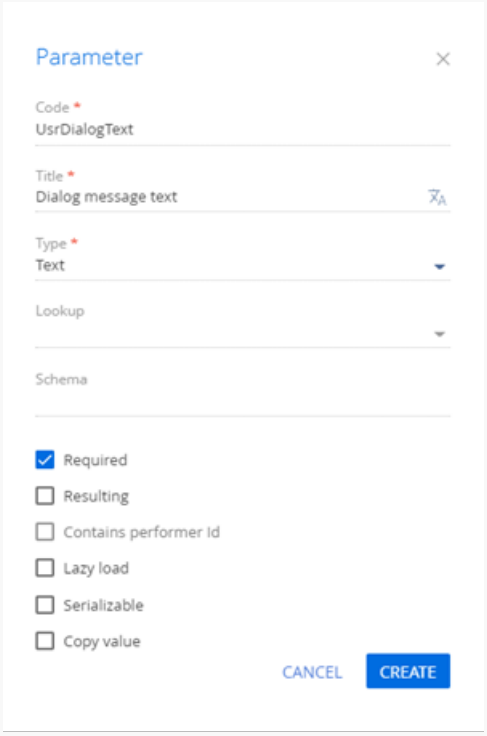
2. Добавить параметры действия процесса

1. Добавьте параметр, который выводит пользователю сообщение.

   a. В узле [*Параметры*] ([*Parameters*]) нажмите кнопку +.

   b. Заполните **свойства параметра**.

   - [*Код*] ([*Code*]) — "`UsrDialogText`".
   - [*Название*] ([*Title*]) — "`Текст в диалоговом окне`" ("`Dialog message text`").
   - [*Тип*] ([*Type*]) — выберите "`Строка`" ("`Text`").
   - Установите признак [*Обязательный для заполнения*] ([*Required*]).

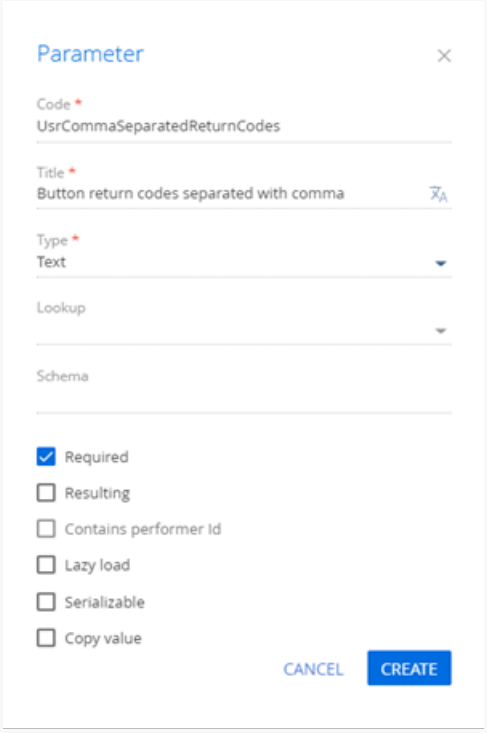


2. Добавьте параметр, который из процесса в элемент передает названия кнопок, которые необходимо отобразить пользователю. Параметр в строке принимает коды кнопок, которые разделены запятой.

   a. В узле [*Параметры*] ([*Parameters*]) нажмите кнопку +.

   b. Заполните **свойства параметра**.

   - [*Код*] ([*Code*]) — "`UsrCommaSeparatedReturnCodes`".
   - [*Название*] ([*Title*]) — "`Коды возврата кнопок через запятую`" ("`Button return codes separated with comma`").
   - [*Тип*] ([*Type*]) — выберите "`Строка`" ("`Text`").
   - Установите признак [*Обязательный для заполнения*] ([*Required*]).

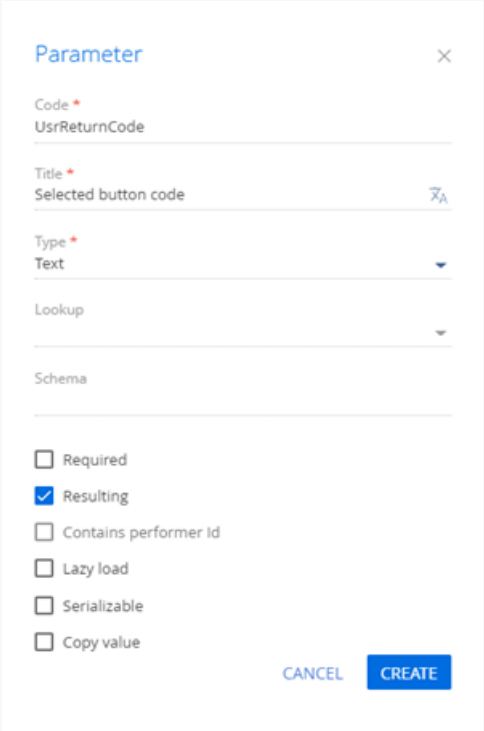


3. Добавьте параметр, который содержит код нажатой кнопки.

   a. В узле [*Параметры*] ([*Parameters*]) нажмите кнопку +.

   b. Заполните **свойства параметра**.

   - [*Код*] ([*Code*]) — "`UsrReturnCode`".
   - [*Название*] ([*Title*]) — "`Код нажатой кнопки`" ("`Selected button code`").
   - [*Тип*] ([*Type*]) — выберите "`Строка`" ("`Text`").
   - Установите признак [*Результирующий*] ([*Resulting*]).



3. Реализовать действие процесса на back-end стороне

1. Настройте логирование процесса.

   a. С помощью директивы `using` добавьте необходимые пространства имен.

```
using global::Common.Logging;
using Terrasoft.Configuration;
```

   b. Реализуйте логику работы логирования и действия процесса.

UsrShowModalPageUserTask

```
namespace Terrasoft.Core.Process.Configuration
{

    using Newtonsoft.Json;
    using Newtonsoft.Json.Linq;
    using System;
    using System.Collections.Generic;
    using System.Collections.ObjectModel;
    using System.Globalization;
    using Terrasoft.Common;
    using Terrasoft.Core;
    using Terrasoft.Core.Configuration;
    using Terrasoft.Core.DB;
```

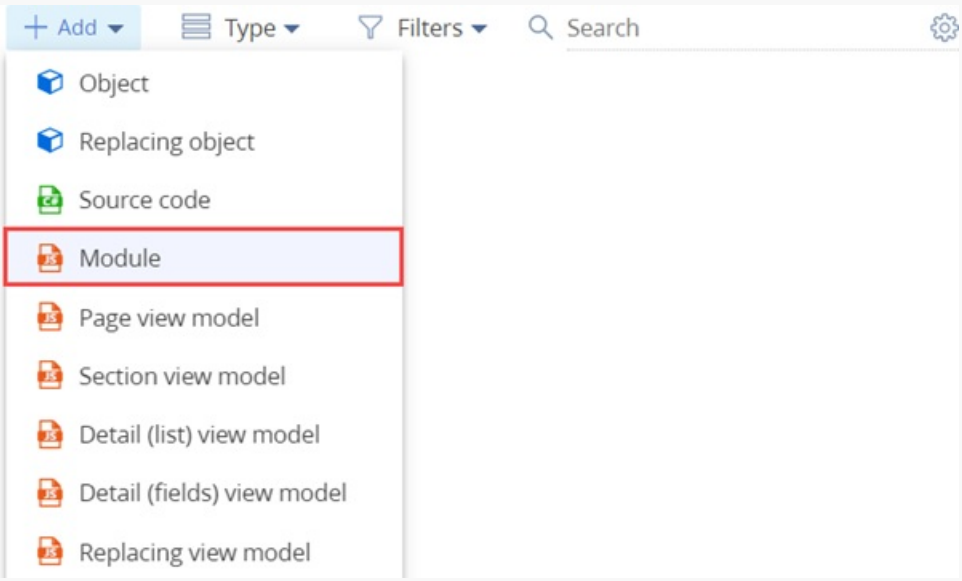
```
using Terrasoft.Core.Entities;
using Terrasoft.Core.Process;
using Terrasoft.UI.WebControls.Controls;

/* Для работы логирования. */
using global::Common.Logging;

/* Для работы инструментов отправки сообщений. */
using Terrasoft.Configuration;

#region Class: UsrShowModalPageUserTask

/// <exclude/>
public partial class UsrShowModalPageUserTask
{
    /* Настройка логирования.
    Создание отдельного логгера. Результаты логирования записываются в файл Common.log
    private static readonly ILog _log = LogManager.GetLogger("UsrShowModalPageUserTask"

    /* Опредение отправителя со стороны back-end. */
    private const string MessageSender = "UsrShowModalPageUserTask";

    #region Methods: Protected

    /* Бизнес-логика действия процесса. */
    protected override bool InternalExecute(ProcessExecutingContext context) {

        /* Вывод информационного сообщения в логи. */
        _log.InfoFormat("UserTask works well. UsrDialogText = {0}, UsrCommaSeparatedRet
        /* Формирование сообщения. */
        var messageData = new
        {
            /* Текст в диалоговом окне. */
            UsrDialogText = UsrDialogText,
            /* Коды возврата кнопок через запятую. */
            UsrCommaSeparatedReturnCodes = UsrCommaSeparatedReturnCodes,
            /* Служебный параметр. Уникальный идентификатор экземпляра элемента внутри
            procElUId = UId
        };
        /* Сериализация объекта тела сообщения. */
        string messageBody = JsonConvert.SerializeObject(messageData);
        /* Тело сообщения, которое отправляется из back-end во front-end часть. */
        MsgChannelUtilities.PostMessage(UserConnection, MessageSender, messageBody);
        return false;
    }

    #endregion
```

```
        #region Methods: Public

        public override bool CompleteExecuting(params object[] parameters) {
            return base.CompleteExecuting(parameters);
        }

        public override void CancelExecuting(params object[] parameters) {
            base.CancelExecuting(parameters);
        }

        public override string GetExecutionData() {
            return string.Empty;
        }

        public override ProcessElementNotification GetNotificationData() {
            return base.GetNotificationData();
        }

        #endregion

    }

    #endregion

}
```

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

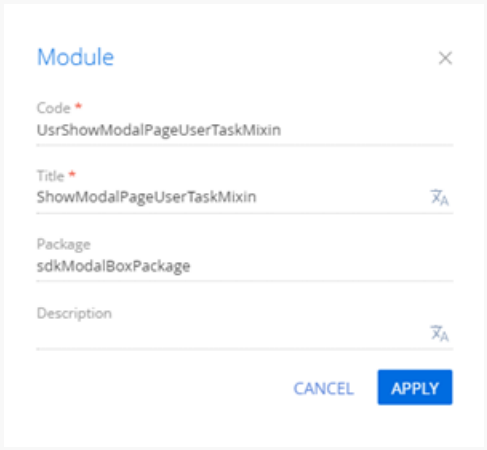
4. Реализовать обработку действия процесса на front-end стороне

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. Заполните **свойства модуля**.

   - [*Код*] ([*Code*]) — "`UsrShowModalPageUserTaskMixin`".
   - [*Заголовок*] ([*Title*]) — "`ShowModalPageUserTaskMixin`".



4. В дизайнере модуля добавьте исходный код.

`UsrShowModalPageUserTaskMixin`

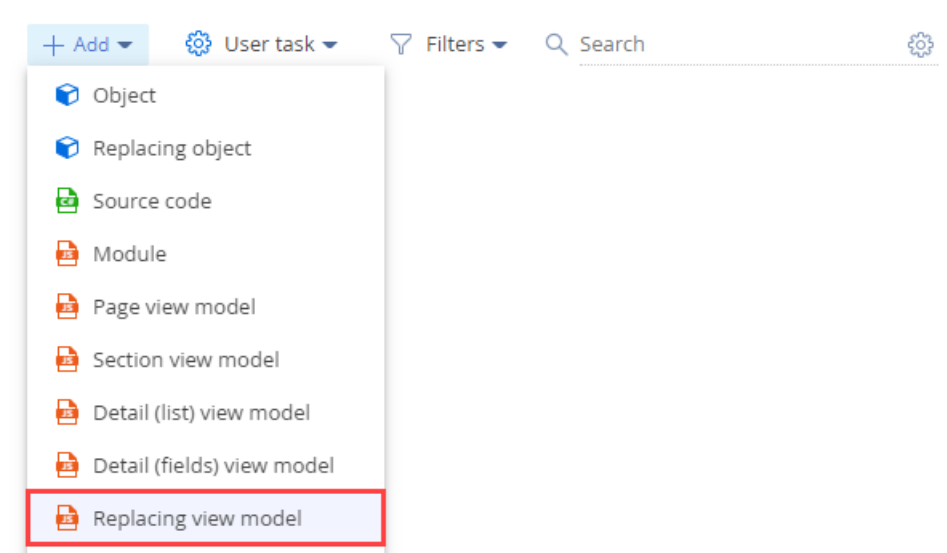
```
define("UsrShowModalPageUserTaskMixin", [],
    function() {
        Ext.define("Terrasoft.configuration.mixins.ShowModalPageUserTaskMixin", {
            alternateClassName: "Terrasoft.ShowModalPageUserTaskMixin",

            /* Подписаться на сообщения канала WebSocket. */
            UsrSubscribeForShowModalWindowServerChannelMessage: function() {
                this.Terrasoft.ServerChannel.on(this.Terrasoft.EventName.ON_MESSAGE, this.Usr
            },
            /* Отписаться от сообщений канала WebSocket. */
```

```
            UsrUnsubscribeForShowModalWindowServerChannelMessage: function() {
                this.Terrasoft.ServerChannel.un(this.Terrasoft.EventName.ON_MESSAGE, this.Usr
            },

            UsrShowModalPageUserTask_procElUId: "",

            /* Функция, которая обрабатывает WebSocket сообщения. */
            UsrServerChannelMessageHandler: function(scope, message) {
                if (!message) {
                    return;
                }
                /* Определение сообщения от отправителя UsrShowModalPageUserTask. */
                if (message.Header && message.Header.Sender !== "UsrShowModalPageUserTask") {
                    return;
                }
                if (message.Body) {
                    var bodyData = this.Ext.decode(message.Body);
                    var UsrDialogText = bodyData.UsrDialogText;
                    var UsrCommaSeparatedReturnCodes = bodyData.UsrCommaSeparatedReturnCodes;

                    /* Получение и сохранение кодов кнопок в виде массива. */
                    var returnCodesArray = UsrCommaSeparatedReturnCodes.split(",");
                    var procElUId = bodyData.procElUId;
                    UsrShowModalPageUserTask_procElUId = procElUId;

                    /* Отобразить диалоговое окно. */
                    this.showConfirmationDialog(UsrDialogText, this.UsrGetConfirmationResult,
                }
            },
            /* Обработка результата выбора кода кнопки. */
            UsrGetConfirmationResult: function(returnCode) {
                this.console.log("UsrGetConfirmationResult: returnCode = " + returnCode + " U

                var procElUId = UsrShowModalPageUserTask_procElUId;

                /* Создание POST-запроса. */
                this.Terrasoft.AjaxProvider.request({
                    url: "../ServiceModel/ProcessEngineService.svc/" + procElUId + "/Complete
                    method: "POST",
                    scope: this,
                    callback: function(request, success, response) {
                    }
                });
            }
        });
    });
```

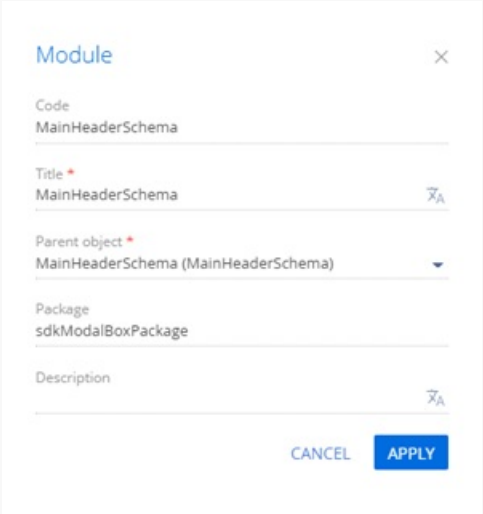
5. На панели инструментов дизайнера нажмите [*Сохранить*] ([*Save*]).

5. Создать замещающую модель представления контейнера

1. Перейдите в раздел [*Конфигурация*] ([*Configuration*]) и выберите пользовательский пакет, в который будет добавлена схема.
2. На панели инструментов реестра раздела нажмите [*Добавить*] —> [*Замещающая модель представления*] ([*Add*] —> [*Replacing view model*]).



3. Заполните **свойства модуля**.
   - [*Код*] ([*Code*]) — "`MainHeaderSchema`".
   - [*Заголовок*] ([*Title*]) — "`MainHeaderSchema`".
   - [*Родительский объект*] ([*Parent object*]) — выберите "`MainHeaderSchema`".



4. В дизайнере модуля добавьте исходный код.

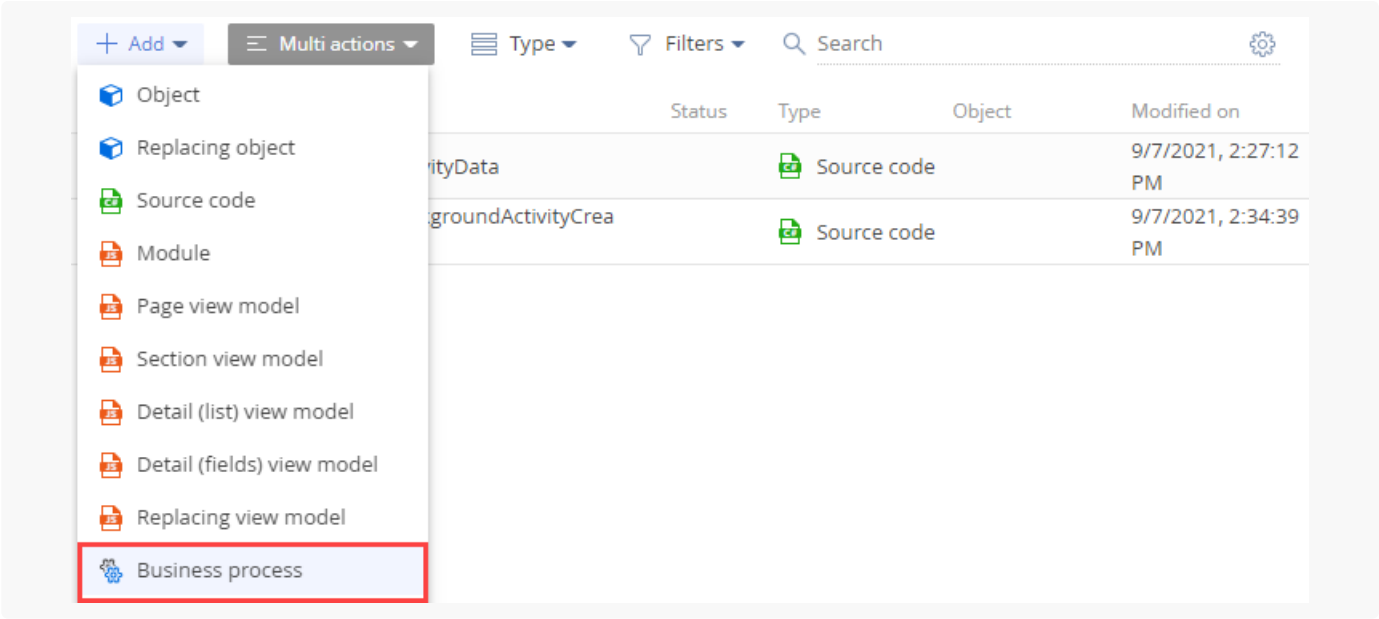
MainHeaderSchema

```
define("MainHeaderSchema", ["UsrShowModalPageUserTaskMixin"], function(){
    return {
        mixins: {
            ShowModalPageUserTaskMixin: "Terrasoft.ShowModalPageUserTaskMixin"
        },
        methods: {
            init: function() {
                this.callParent(arguments);
                this.UsrSubscribeForShowModalWindowServerChannelMessages();
            },
            destroy: function() {
                this.callParent(arguments);
                this.UsrUnsubscribeForShowModalWindowServerChannelMessages();
            }
        }
    };
});
```

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

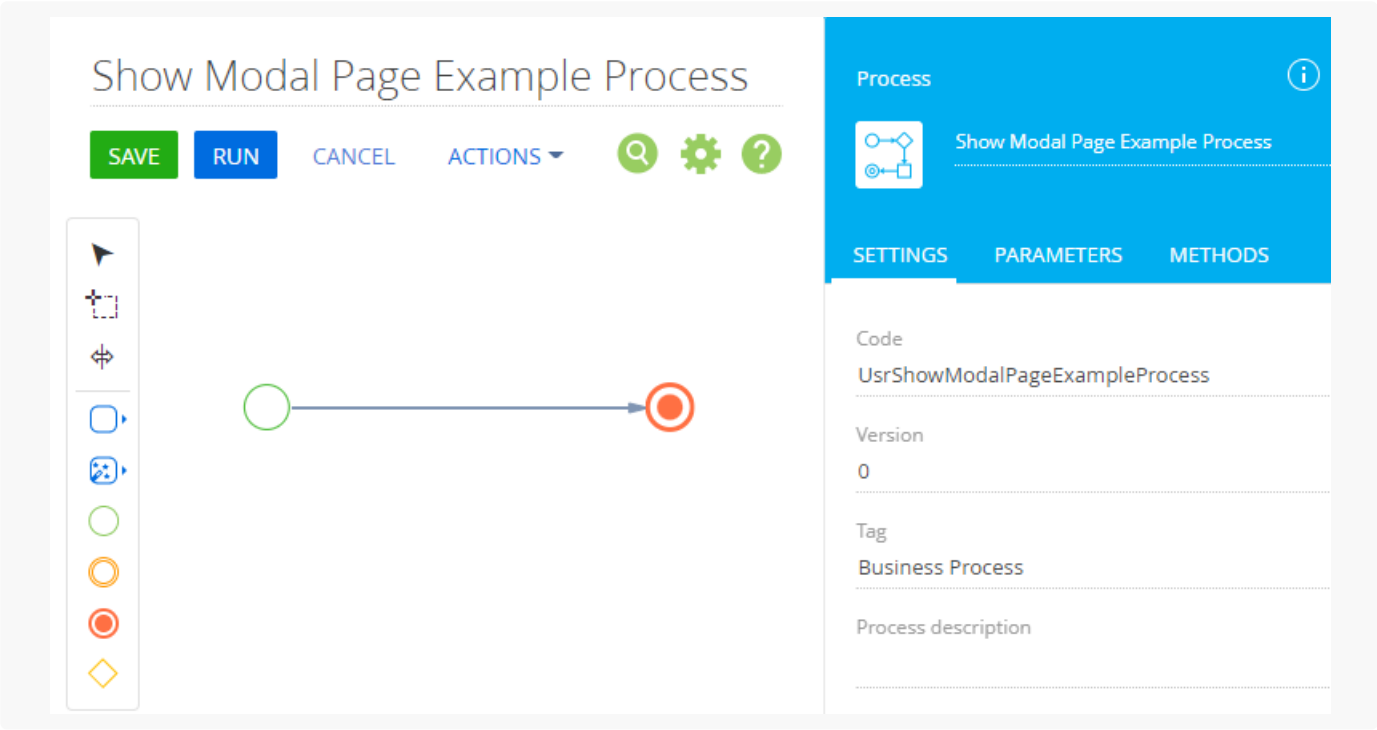
6. Создать бизнес-процесс отображения модального окна

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**.

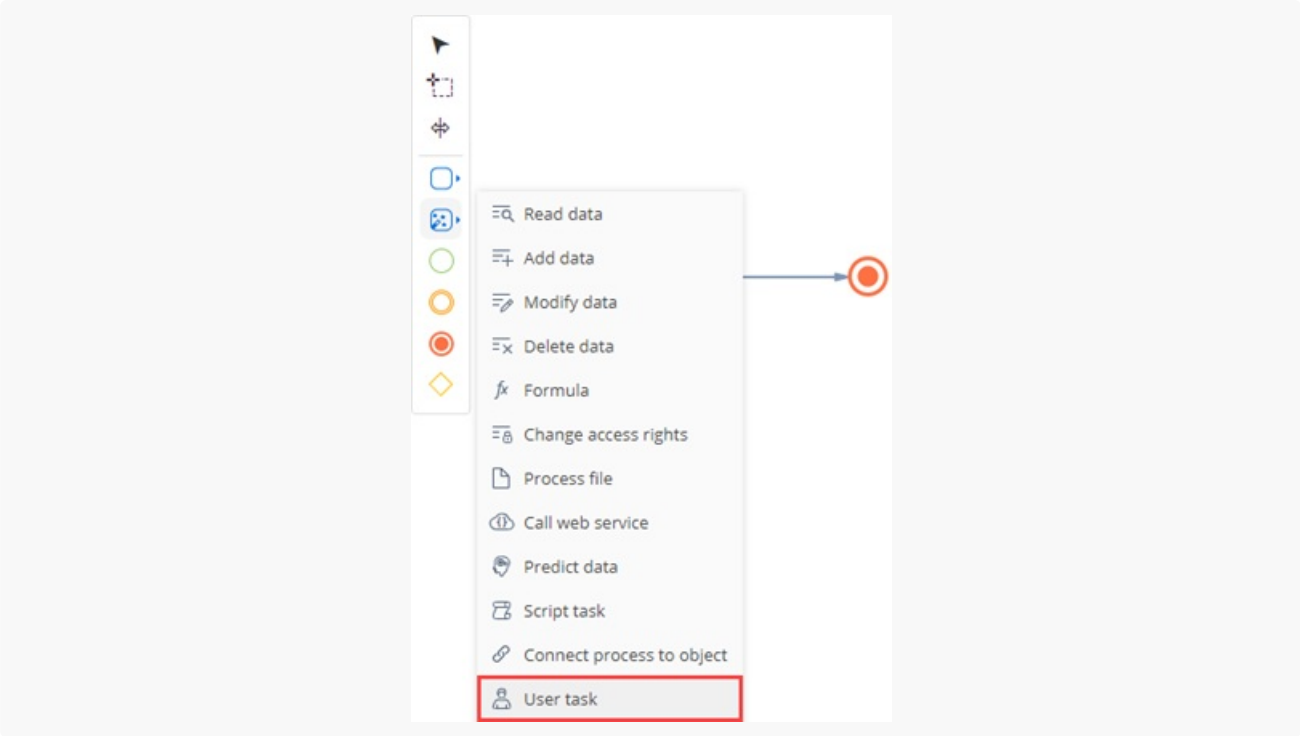
   - На панели настройки элементов заполните свойство [*Заголовок*] ([*Title*]) — "`Show Modal Page Example Process`".
   - На вкладке [*Настройки*] ([*Settings*]) панели настройки элементов заполните свойство [*Имя*] ([*Code*]) — "`UsrShowModalPageExampleProcess`".



4. Реализуйте бизнес-процесс.

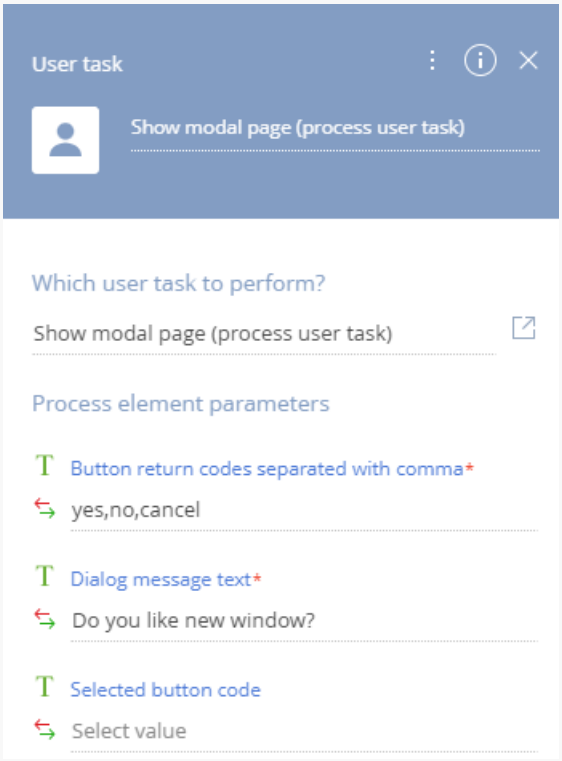
   a. Добавьте действие процесса.

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



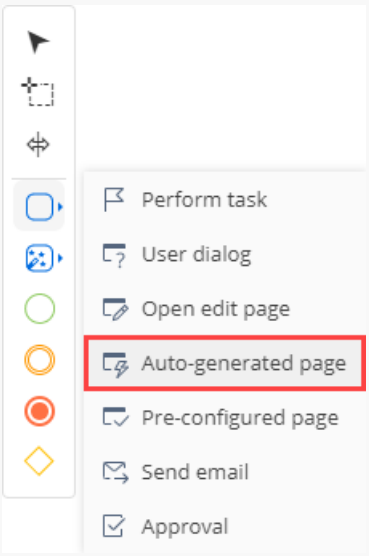
b. Заполните **свойства действия процесса**.

- [*Какое пользовательское действие выполнить?*] ([*Which user task to perform?*]) — выберите "`Показать модальное окно (действие процесса)`" ("`Show modal page (process user task)`").
- Заполните значения параметров действия процесса.
  - [*Текст в диалоговом окне*] ([*Dialog message text*]) — "`Вам нравится новое окно?`" ("`Do you like new window?`").
  - [*Коды возврата кнопок через запятую*] ([*Button return codes separated with comma*]) — "`yes,no,cancel`".



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**.

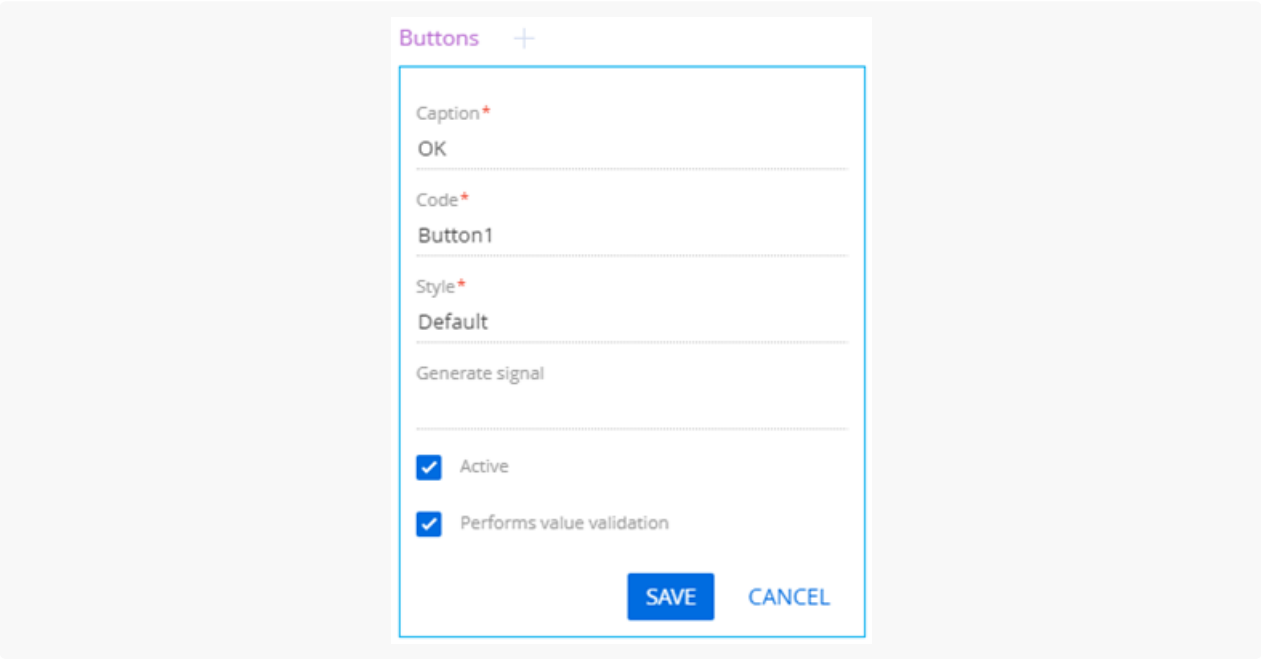
   - [*Заголовок*] ([*Title*]) — "YES".
   - [*Название страницы*] ([*Page title*]) — "Нажато YES" ("Pressed YES").

e. Добавьте кнопку.

a. В блоке [*Кнопки*] ([*Buttons*]) нажмите кнопку +.

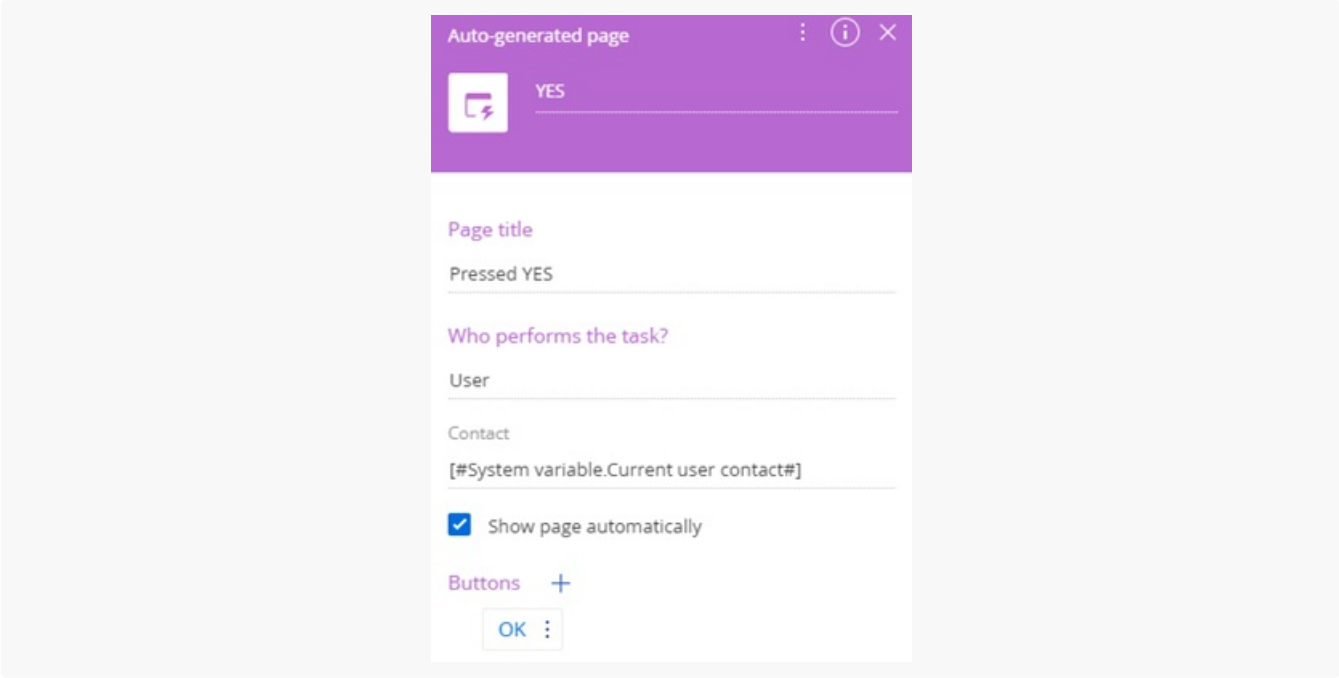
b. Заполните **свойства кнопки**.

- [*Название*] ([*Caption*]) — "ОК".



d. Сохраните изменения.

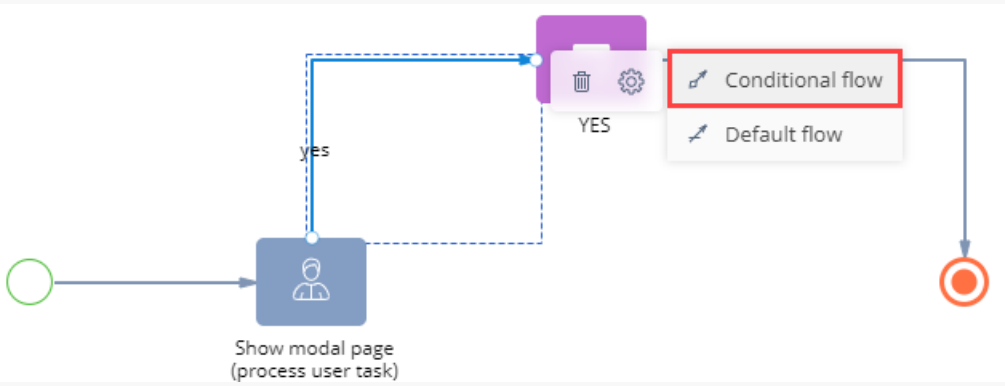
Настройки автогенерируемой страницы представлены на рисунке ниже.



c. Настройте условный поток.

a. В меню действия процесса выберите [*Добавить поток*] ([*Add flow*]) и соедините действие процесса с автогенерируемой страницей.

b. Трансформируйте поток управления в условный поток. Для этого нажмите кнопку и в меню потока выберите [*Условный поток*] ([*Conditional flow*]).



c. Заполните **свойства условного потока**.

- [*Заголовок*] ([*Title*]) — "yes".

e. Настройте условия перехода.

a. На панели настройки элементов в свойстве [*Условие перехода*] ([*Condition to move down the flow*]) нажмите кнопку .

b. Выберите элемент процесса "Показать модальное окно (действие процесса)" ("Show modal page (process user task)").

c. Двойным кликом выберите параметр процесса "Код нажатой кнопки" ("Selected button code").

d. Задайте формулу для параметра.

```
[#Show modal page (process user task).Selected button code#] == "yes"
```

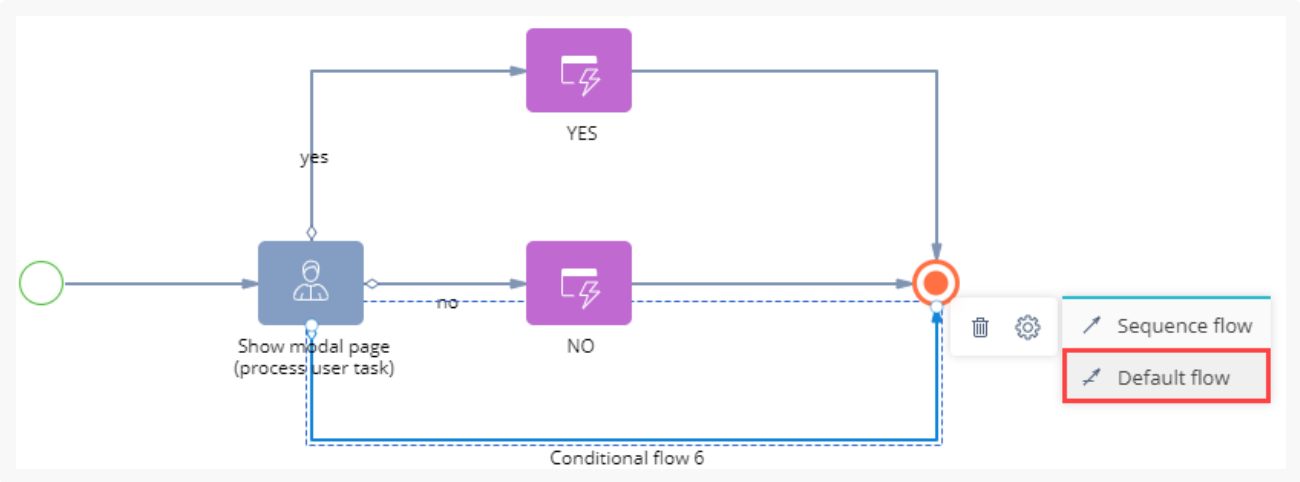
e. Сохраните изменения.

d. Аналогично добавьте автогенерируемую страницу `NO` с соответствующим условным потоком `no`.

e. Настройте поток по умолчанию.

a. В меню действия процесса выберите [*Добавить поток*] ([*Add flow*]) и соедините действие процесса с завершающим событием [*Останов*] ([*Terminate*]).

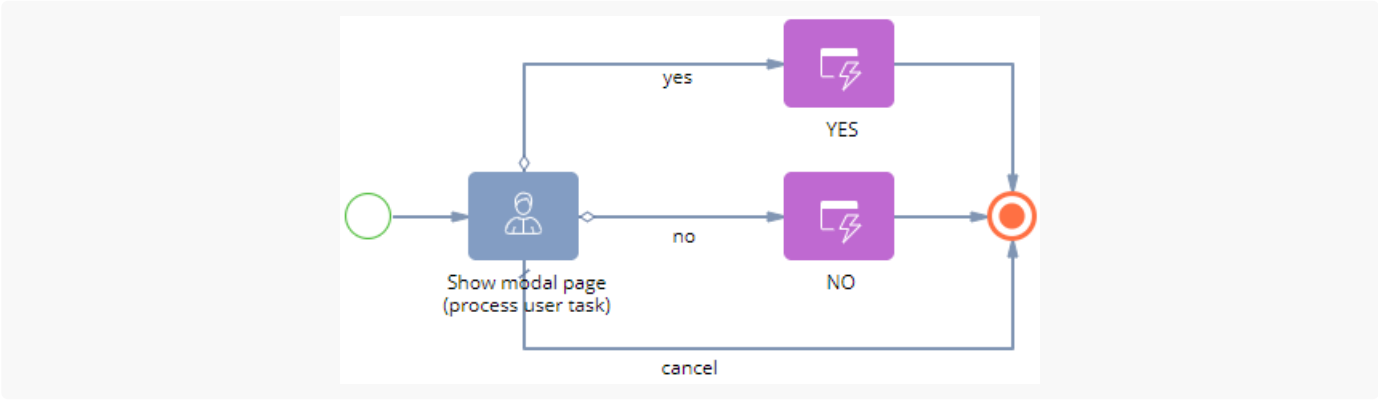
b. Трансформируйте условный поток в поток по умолчанию. Для этого нажмите кнопку и в меню потока выберите [*Поток по умолчанию*] ([*Default flow*]).



c. Заполните **свойства потока по умолчанию**.

- [*Заголовок*] ([*Title*]) — "cancel".

Бизнес-процесс представлен на рисунке ниже.



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

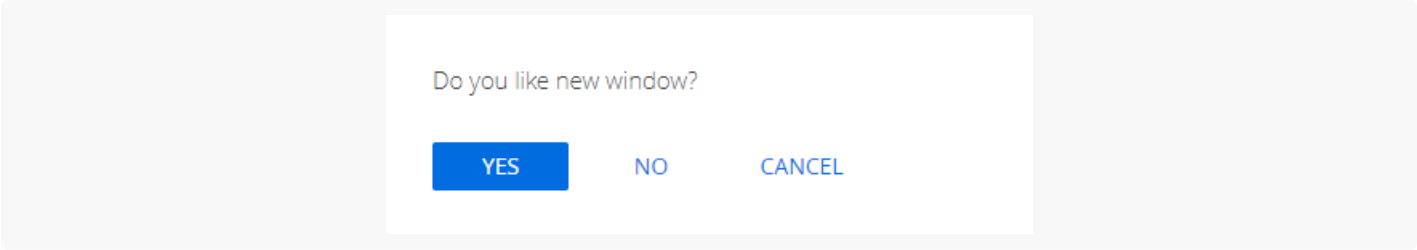
Результат выполнения примера

Бизнес-процесс `Show Modal Page Example Process` доступен для запуска в любом разделе приложения Creatio. Например, запустим бизнес-процесс из раздела [*Контакты*] ([*Contacts*]).

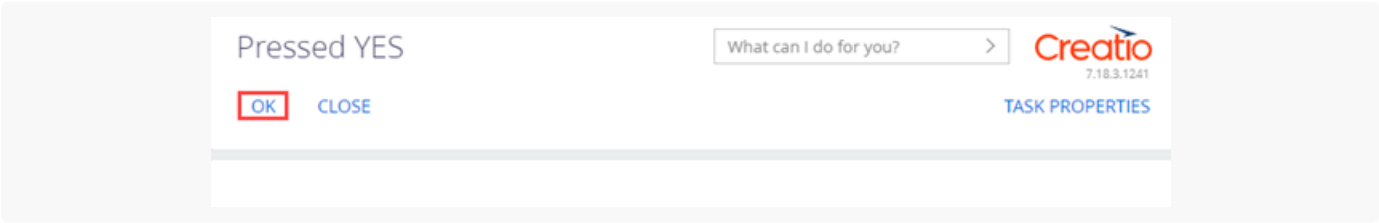
Чтобы **запустить бизнес-процесс** `Show Modal Page Example Process` из раздела [*Контакты*] ([*Contacts*]):

1. Перейдите в раздел [*Контакты*] ([*Contacts*]).
2. На панели разделов нажмите кнопку  .
3. Выберите бизнес-процесс `Show Modal Page Example Process` и нажмите [*Запустить*] ([*Run*]).

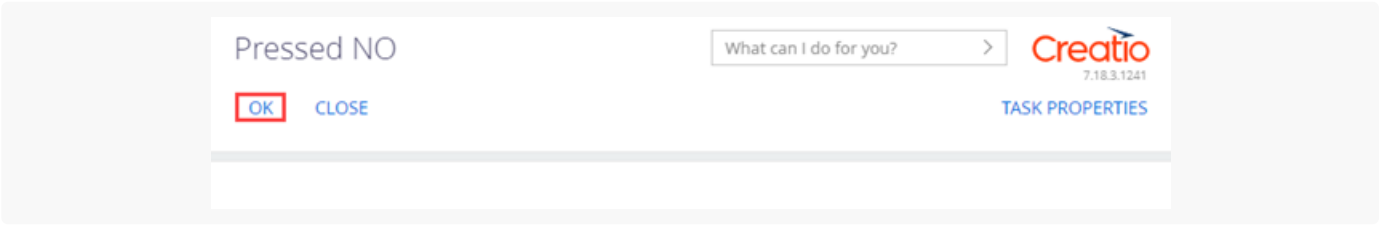
В результате выполнения бизнес-процесса `Show Modal Page Example Process` в разделе [*Контакты*] ([*Contacts*]) отображается модальное окно.



- При нажатии на кнопку [*Yes*] отображается страница [*Нажато YES*] ([*Pressed YES*]) с кнопкой [*OK*].



- При нажатии на кнопку [*No*] отображается страница [*Нажато NO*] ([*Pressed NO*]) с кнопкой [*OK*].



- При нажатии на кнопку [*Cancel*] модальное окно закрывается.

Результаты логирования запросов отображаются в файле `Common.log` корневой папки приложения. Используя инструменты разработчика в браузере, можно отследить WebSocket-сообщения, которые переданы из back-end части во front-end часть.