

# Разработка первого приложения

Шаг 4. Реализуйте автозаполнение расписания

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>Шаг 4. Реализуйте автозаполнение расписания</b>	<b>4</b>
Создать бизнес-процесс	4
Создать замещающую модель представления	13
Отредактировать исходный код страницы	15

## Шаг 4. Реализуйте автозаполнение расписания

### Основы

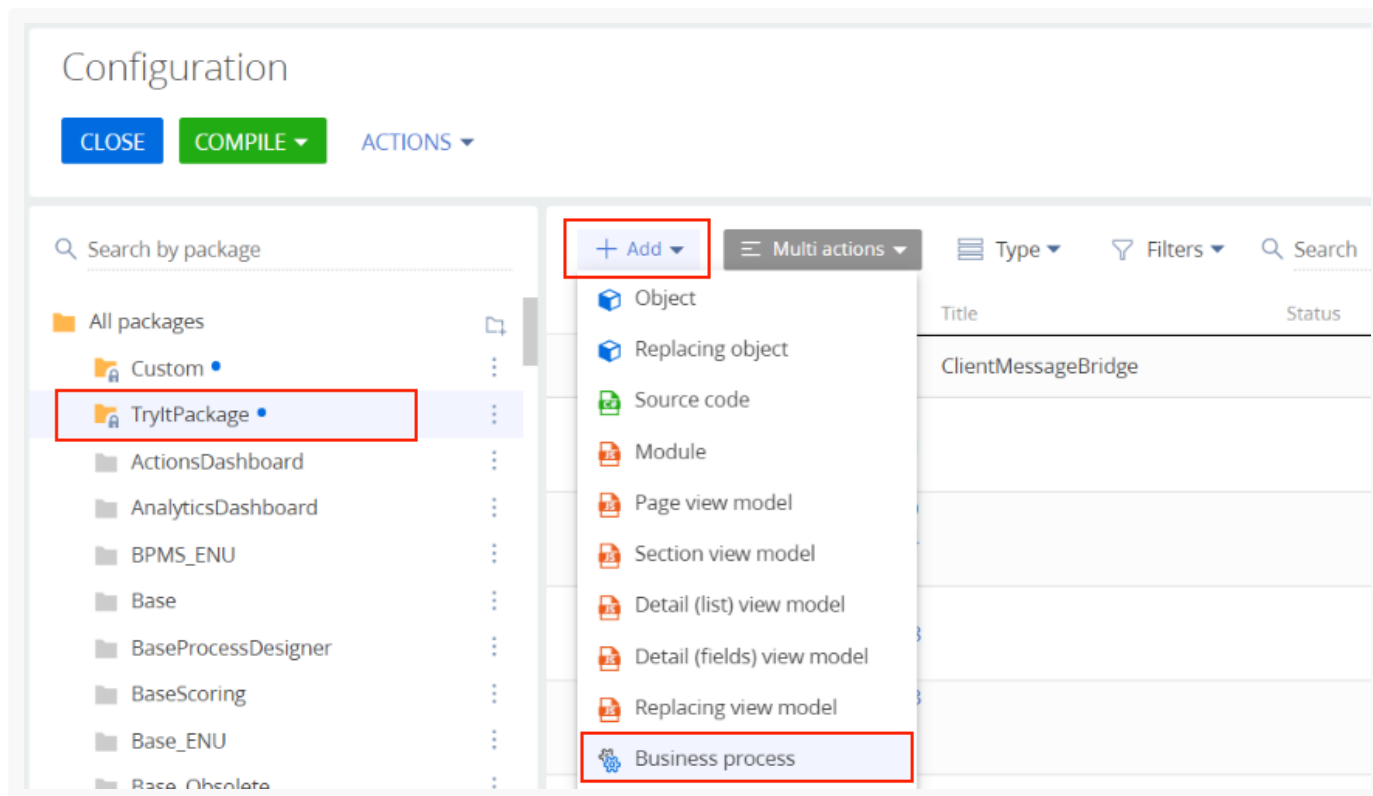
На [предыдущем шаге](#) мы добавили к странице проверку соответствия текущего количества активных ежедневных секций количеству залов для групповых занятий.

Теперь реализуем автоматическое заполнение расписания групповых занятий — добавим на страницу действие, которое будет добавлять в расписание 4 новых записи.

Для этого создадим [бизнес-процесс](#), который будет записывать в таблицу базы данных необходимые данные и возвращать в схему страницы сообщение о том, что записи добавлены.

### Создать бизнес-процесс

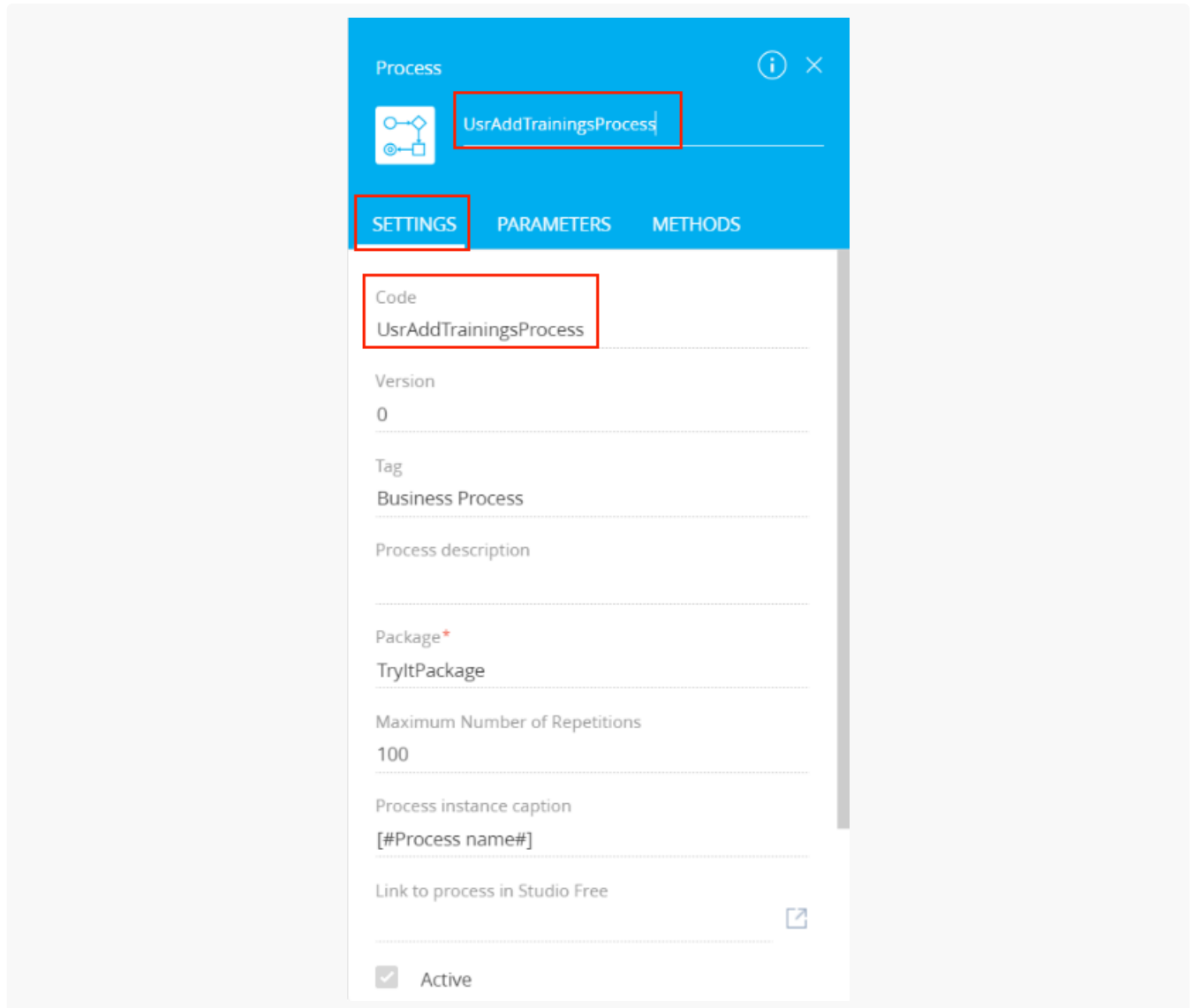
1. [Перейдите в раздел \[ Конфигурация \]](#) ([ Configuration ]).
2. В списке пакетов выберите [пакет](#) "TryItPackage".
3. На панели инструментов рабочей области нажмите кнопку [ [Добавить](#) ] ([ Add ]) и выберите в списке тип конфигурационного элемента [ [Бизнес процесс](#) ] ([ Business process ]).



4. **Заполните поля свойств** процесса значениями:

- [ [Название](#) ] ([ Name ]) — "UsrAddTrainingsProcess".

- [ Код ] ([ Code ]) — "UsrAddTrainingsProcess".



5. На вкладку [ Параметры ] ([ Parameters ]) добавьте входящие параметры и параметры, необходимые для работы процесса:

[ Название ] ([ Title ])	[ Код ] ([ Code ])	[ Тип данных ] ([ Data type ])	[ Направление ] ([ Direction ])	[ Значение ] ([ Value ])
Process Schema Periodicity	Process Schema Periodicity	Строка (500 символов) (Text (500 characters))	Входящий (Input)	
NewDate	NewDate	Дата/Время (Date/Time)	Двунаправленный (Bidirectional)	
Number	Number	Целое число (Integer)	Двунаправленный (Bidirectional)	4
Process SchemaId	Process SchemaId	Уникальный идентификатор (Unique identifier)	Входящий (Input)	
Process SchemaCoach	Process SchemaCoach	Уникальный идентификатор (Unique identifier)	Входящий (Input)	

Process

UsrAddTrainingsProcess

SETTINGS

PARAMETERS

METHODS

ADD PARAMETER

Id

ProcessSchemaCoach

→ Select value

Id

ProcessSchemaId

→ Select value

T

ProcessSchemaPeriodicity

→ Select value

NewDate

↔ Select value

123

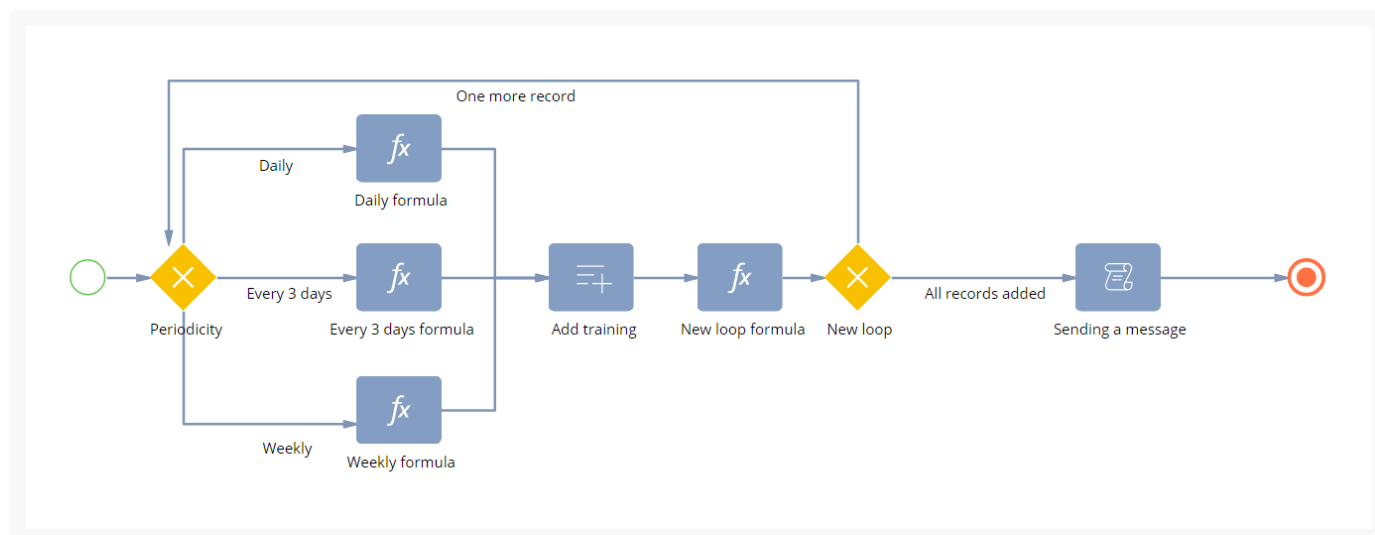
Number

↔ 4

6. Разместите в рабочей области процесса элементы:

- [ *Исключающее ИЛИ* ] ([ *Exclusive gateway (OR)* ]).
- [ *Формула* ] ([ *Formula* ]).
- [ *Добавить данные* ] ([ *Add data* ]).
- [ *Задание-сценарий* ] ([ *Script task* ]).

Соедините элементы стрелками, как показано на рисунке ниже.



Процесс выполняется следующим образом:

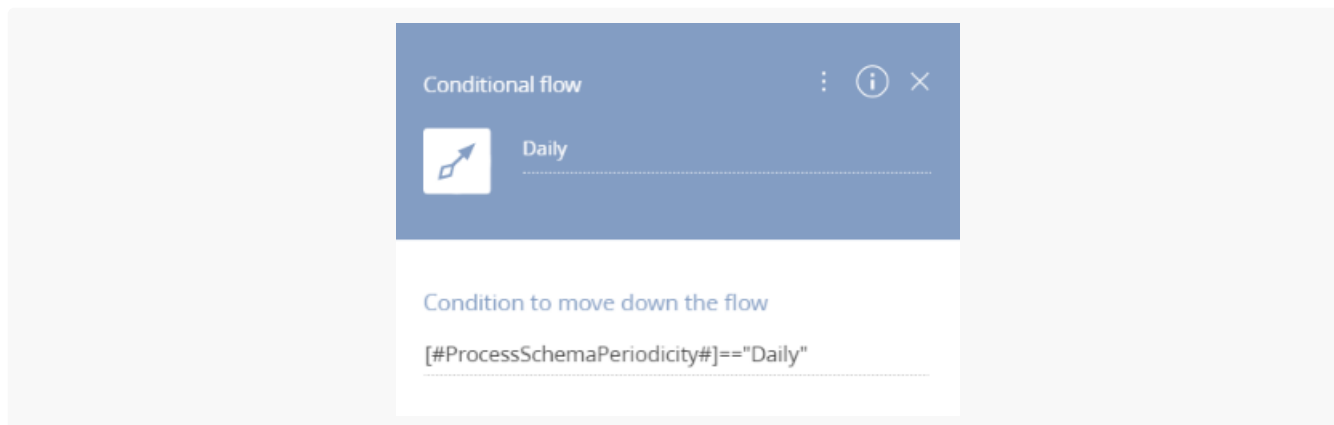
- Логический оператор [ *Исключающее ИЛИ* ] ([ *Exclusive gateway (OR)* ] ) "Periodicity" — проверяет периодичность секции, для которой нужно заполнить расписание, и разветвляет процесс на следующем шаге в зависимости от периодичности.
- Элементы [ *Формула* ] ([ *Formula* ] ) — процесс выполняет расчет даты занятия, в зависимости от периодичности секции. Дата записывается в параметр процесса `NewDate`.
- Элемент [ *Добавить данные* ] ([ *Add data* ] ) — процесс записывает новые данные группового занятия в таблицу базы данных.
- Элемент [ *Формула* ] ([ *Formula* ] ) "New loop formula" — процесс выполняет расчет количества оставшихся для заполнения записей. По условию примера, необходимо заполнить и записать 4 новых записи. Результат записывается в параметр процесса `Number`.
- Логический оператор [ *Исключающее ИЛИ* ] ([ *Exclusive gateway (OR)* ] ) "New loop" — проверяет значение параметра процесса `Number`. Если оно больше 0, то выполняет повторение предыдущих шагов. Если значение параметра равно 0, то выполняется переход к следующему шагу.
- Элемент [ *Задание-сценарий* ] ([ *Script task* ] ) — процесс выполняет публикацию сообщения о том, что новые записи групповых занятий добавлены.

7. Настройте элементы бизнес-процесса.

**[ *Исключающее ИЛИ* ] ([ *Exclusive gateway (OR)* ] ) "Periodicity".**

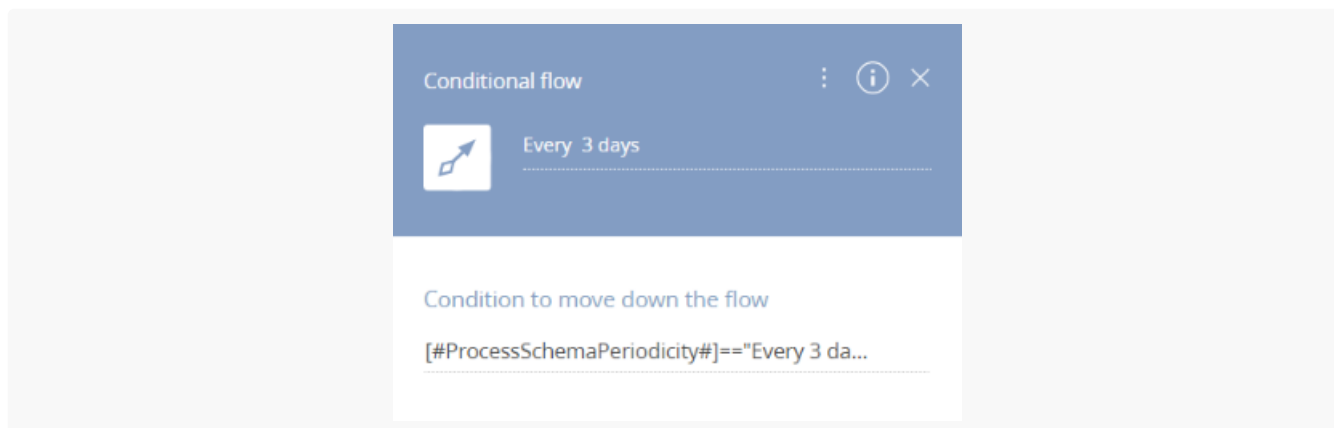
a. [ *Условный поток* ] ([ *Conditional flow* ] ) "Daily":

- [ Условие перехода ]([ Condition to move down the flow ]) — `[#ProcessSchemaPeriodicity#]=="Daily"`.



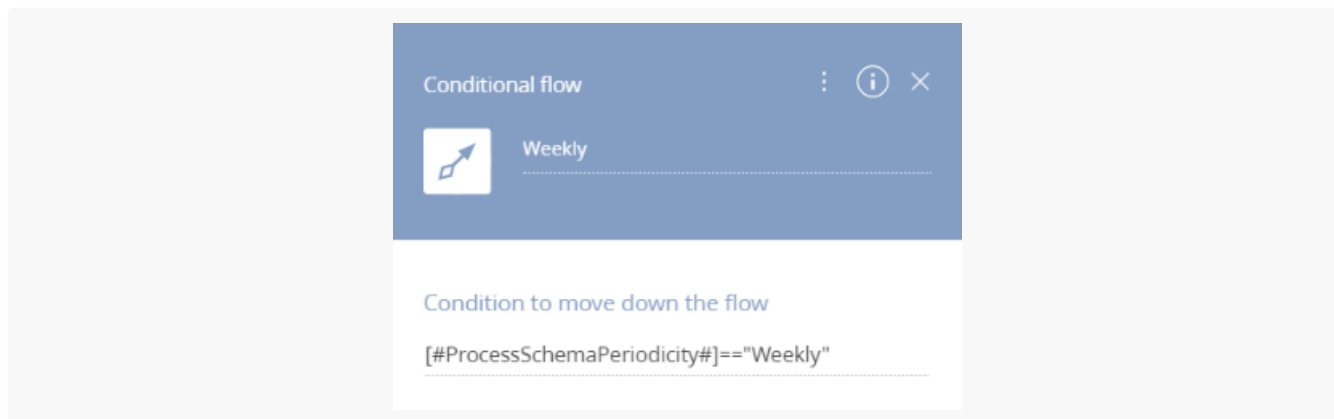
с. [ Условный поток ]([ Conditional flow ]) "Every 3 days":

- [ Условие перехода ]([ Condition to move down the flow ]) — `[#ProcessSchemaPeriodicity#]=="Every 3 days"`.



е. [ Условный поток ]([ Conditional flow ]) "Weekly":

- [ Условие перехода ]([ Condition to move down the flow ]) — `[#ProcessSchemaPeriodicity#]=="Weekly"`.



[ Формула ]([ Formula ]) "Daily formula":

- [ В какой параметр передать значение формулы? ]([ Which parameter to set the formula value to? ]) —



"NewDate".

- [ *Значение формулы* ]([ *Formula value* ]) — `DateTime.Now.AddDays(4-[#Number#])` .

The screenshot shows a 'Formula' dialog box with a blue header. Below the header, there is a section titled 'Daily formula' with an 'fx' icon. The dialog contains two input fields: 'Which parameter to set the formula value to?' with 'NewDate' entered, and 'Formula value' with the formula `DateTime.Now.AddDays(4-[#Number#])` entered.

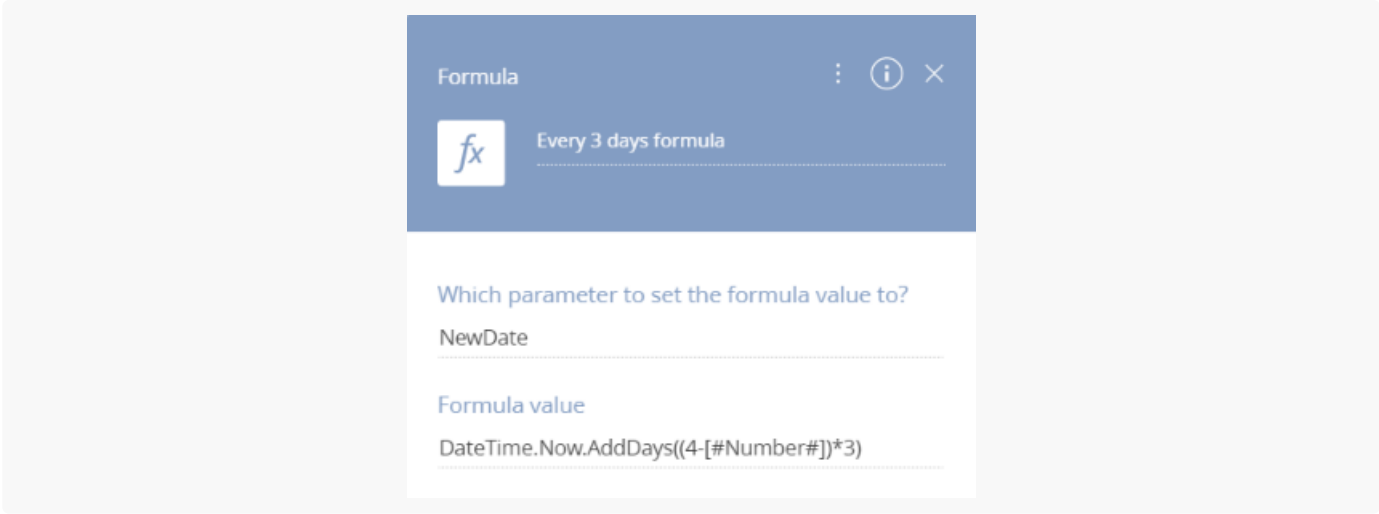
**[ Формула ] ([ *Formula* ]) "Every 3 days formula":**

- [ *В какой параметр передать значение формулы?* ]([ *Which parameter to set the formula value to?* ]) — "NewDate".
- [ *Значение формулы* ]([ *Formula value* ]) — `DateTime.Now.AddDays((4-[#Number#])*3)` .

The screenshot shows a 'Formula' dialog box with a blue header. Below the header, there is a section titled 'Every 3 days formula' with an 'fx' icon. The dialog contains two input fields: 'Which parameter to set the formula value to?' with 'NewDate' entered, and 'Formula value' with the formula `DateTime.Now.AddDays((4-[#Number#])*3)` entered.

**[ Формула ] ([ *Formula* ]) "Weekly formula":**

- [ *В какой параметр передать значение формулы?* ]([ *Which parameter to set the formula value to?* ]) — "NewDate".
- [ *Значение формулы* ]([ *Formula value* ]) — `DateTime.Now.AddDays((4-[#Number#])*7)` .



**[ Добавить данные ] ([ Add data ]) "Add training":**

- [ В какой объект добавить данные? ]([ Which object to add data to? ]) — "Group training".
- [ Какой режим добавления данных? ]([ What is the data adding mode? ]) — Добавить одну запись (Add one record).
- [ Какие значения колонок установить? ]([ Which column values to set? ])

Колонка объекта	Тип добавляемых данных	Значение
[Group sections]	Параметр процесса	ProcessSchemaId
[Training date]	Параметр процесса	NewDate
[Training status]	Значение справочника	Planned
[Training time]	Выбор времени	9:00

Add data

Add training

Which object to add data to?

Group training

What is the data adding mode?

Add one record

Which column values to set?

Group sections

[#ProcessSchemaId#]

Training date

[#NewDate#]

Training status

[#Lookup.Training status.Planned.ddf38dae-;

Coach

[#ProcessSchemaCoach#]

Training time

[#Time value.9:00 AM#]

+ Add field

**[ Формула ] ([ Formula ]) "New loop formula":**

- [ В какой параметр передать значение формулы? ]([ Which parameter to set the formula value to? ]) — "Number".
- [ Значение формулы ]([ Formula value ]) — [#Number#]-1 .

**[ Исключающее ИЛИ ] ([ Exclusive gateway (OR) ]) "New loop".**

a. [ Условный поток ] ([ Conditional flow ]) "One more record":

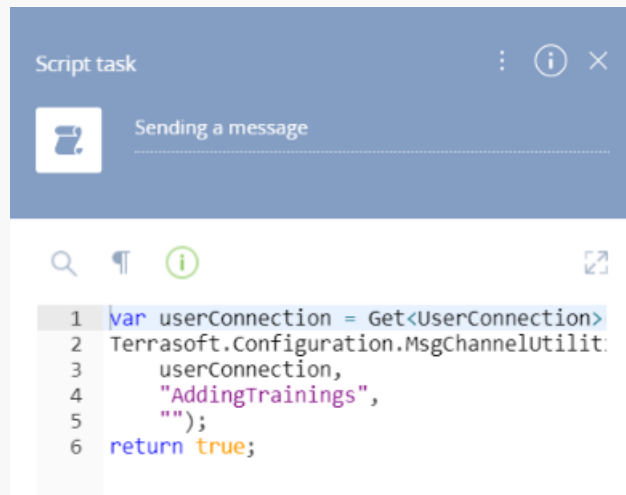
- [ Условие перехода ]([ Condition to move down the flow ]) — `[#Number#]>=1`.

c. [ Условный поток ] ([ Conditional flow ]) "All records added":

- [ Условие перехода ]([ Condition to move down the flow ]) — `[#Number#]<1`.

**[ Задание-сценарий ] ([ Script task ]) "Sending a message":**

```
var userConnection = Get<UserConnection>("UserConnection");
Terrasoft.Configuration.MsgChannelUtilities.PostMessage(
    userConnection,
    "AddingTrainings",
    "");
return true;
```



8. Сохраните процесс.

## Создать замещающую модель представления

Для реализации обмена сообщениями между процессом и модулем страницы создайте замещающую модель представления базовой схемы `ClientMessageBridge`. Эта схема реализует транслирование сообщений, полученных по [WebSocket](#), подписчикам внутри Creatio.

1. [Перейдите в раздел \[ Конфигурация \]](#) ([ *Configuration* ]).
2. В списке пакетов выберите [пакет](#) "TryItPackage".
3. На панели инструментов рабочей области нажмите кнопку [ *Добавить* ] ([ *Add* ]) и выберите в списке тип конфигурационного элемента [ *Замещающая модель представления* ] ([ *Replacing view model* ]).
4. Заполните поле [ *Родительский объект* ] ([ *Parent object* ]) модели представления значением "ClientMessageBridge (ClientMessageBridge)". Остальные поля будут заполнены автоматически.

Module

Code  
ClientMessageBridge

Title \*  
ClientMessageBridge

Parent object \*  
ClientMessageBridge (ClientMessageBridge)

Package  
TryItPackage

Description

CANCEL APPLY

5. В дизайнере схем добавьте исходный код.

ClientMessageBridge.js

```
define("ClientMessageBridge", ["ConfigurationConstants"],
function(ConfigurationConstants) {
    return {
        /* Сообщения. */
        messages: {
            /* Имя сообщения. */
            "AddingTrainings": {
                /* Тип сообщения – широковещательное, без указания конкретного подписчика
                "mode": Terrasoft.MessageMode.BROADCAST,
                /* Направление сообщения – публикация. */
                "direction": Terrasoft.MessageDirectionType.PUBLISH
            }
        },
        methods: {
            /* Инициализация схемы. */
            init: function() {
                /* Вызов родительского метода. */
                this.callParent(arguments);
                /* Добавление нового конфигурационного объекта в коллекцию конфигурационн
                this.addMessageConfig({
                    /* Имя сообщения, получаемого по WebSocket. */
                    sender: "AddingTrainings",
                    /* Имя сообщения, с которым оно будет разослано. */
                    messageName: "AddingTrainings"
                });
            }
        }
    };
});
```

```

    }
  }
};
});

```

6. Сохраните схему, нажав кнопку [ *Сохранить* ] ([ *Save* ]).

## Отредактировать исходный код страницы

На страницу групповой секции добавьте действие, которое будет автоматически добавлять новые занятия на деталь “Групповые занятия”.

1. [Перейдите в раздел \[ Конфигурация \]](#) ([ *Configuration* ]).
2. В списке пакетов выберите [пакет](#) "TryItPackage".
3. В результате работы мастеров в пакете уже содержатся схемы разных типов. С помощью сортировки по типу схем выберите для отображения схемы с типом [ *Клиентский модуль* ] ([ *Client module* ]).

Configuration

CLOSE COMPPILE ACTIONS

Search by package

All packages

- Custom
- TryItPackage
- ActionsDashboard
- AnalyticsDashboard
- BPMS\_ENU
- Base
- BaseProcessDesigner

+ Add Multi actions Client module Filters Search

Name	Title	Status	Type	Object	Modified on	Package
ClientMessageBridge	ClientMessageBridge		Client module		4/8/2021, 10:33:07 PM	TryItPackage
UsrGroupSections1Page	Edit page "Group sections"		Client module		4/13/2021, 12:24:56 AM	TryItPackage
UsrGroupSections224446aaSection	Section schema: "Group sections"		Client module		3/25/2021, 6:16:33 PM	TryItPackage
UsrSchema260be8a1Detail	Detail schema: "Group training"		Client module		3/25/2021, 6:03:03 AM	TryItPackage
UsrSchema3ae6a7afPage	Group training		Client module		3/25/2021, 6:06:01 AM	TryItPackage

4. Двойным нажатием мыши откройте схему `UsrGroupSections1Page` и отредактируйте исходный код:

### UsrGroupSections1Page

```

define("UsrGroupSections1Page", ["ProcessModuleUtilities"], function(ProcessModuleUtilities)
return {
  entitySchemaName: "UsrGroupSections",
  messages: {
    /* Сообщение, которое вызывает обновление детали. */
    "AddingTrainings": {
      "mode": Terrasoft.MessageMode.BROADCAST,
      "direction": Terrasoft.MessageDirectionType.SUBSCRIBE
    }
  },
  /* Без изменений. */
  attributes: {
    // ...
  },

```

```

modules: /**SCHEMA_MODULES*/{}/**SCHEMA_MODULES*/,
/* Без изменений. */
details: /**SCHEMA_DETAILS*/{
    // ...
}/**SCHEMA_DETAILS*/,

businessRules: /**SCHEMA_BUSINESS_RULES*/{}/**SCHEMA_BUSINESS_RULES*/,
/* Добавляем новые методы к существующим. */
methods: {

    // ...

    init: function() {
        this.callParent(arguments);
        /* Подписка на сообщение, которое вызывает обновление детали. */
        this.sandbox.subscribe("AddingTrainings", this.updateTrainings, this);
    },
    /* Добавляет действие в меню действий. */
    getActions: function() {
        var actionMenuItems = this.callParent(arguments);
        actionMenuItems.addItem(this.getButtonMenuItem({
            "Caption": {bindTo: "Resources.Strings.AddTrainingsActionCaption"},
            // Определяем метод-обработчик для действия.
            "Click": {bindTo: "getBusinessProcessAddTrainings"},
            "Enabled": true
        }));
        return actionMenuItems;
    },
    /* Вызывает обновление детали на странице записи. */
    updateTrainings: function() {
        this.updateDetail({
            // Код детали.
            "detail": "UsrSchema12c4c6adDetail5a26acfb",
            "reloadAll": true
        });
    },
    /* Метод-обработчик для нового действия в меню. */
    getBusinessProcessAddTrainings: function() {
        /* Получаем необходимые для процесса входящие параметры. */
        var id = this.get("Id");
        var periodicity = this.get("UsrPeriodicity").displayValue;
        var coach = this.get("UsrCoach").value;
        if (!periodicity) {
            return;
        }
        /* Создаем конфигурационный объект для запуска процесса. */
        var args = {
            /* Имя созданного в предыдущих пунктах процесса. */
            sysProcessName: "UsrAddTrainingsProcess",

```



```

/* Входящие параметры процесса. */
parameters: {
    ProcessSchemaId: id,
    ProcessSchemaPeriodicity: periodicity,
    ProcessSchemaCoach: coach
}
};
/* Запуск процесса. */
ProcessModuleUtilities.executeProcess(args);
},
},
dataModels: /**SCHEMA_DATA_MODELS*/{}/**SCHEMA_DATA_MODELS*/,
/* Без изменений. */
diff: /**SCHEMA_DIFF*/[

// ...

]/**SCHEMA_DIFF*/
};
});

```

5. Сохраните схему, нажав на кнопку [ *Сохранить* ] ([ *Save* ]).

В результате мы реализовали автоматическое заполнение расписания групповых занятий. На [следующем шаге](#) необходимо реализовать веб-сервис, который будет предоставлять информацию о количестве занятий в расписании.