

Разработка первого приложения

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

С чего начать	4
Описание задачи	4
Предварительные настройки	4
Создать пакет для разработки	4
Переопределить текущий пакет	6
Шаг 1. Создайте новый раздел	6
Настроить раздел	7
Настроить деталь	13
Шаг 2. Добавьте новые данные	18
Заполнить справочники	18
Создать системную настройку	21
Привязать данные к пакету	23
Шаг 3. Добавьте валидацию страницы	30
Шаг 4. Реализуйте автозаполнение расписания	34
Создать бизнес-процесс	34
Создать замещающую модель представления	44
Отредактировать исходный код страницы	46
Шаг 5. Создайте пользовательский веб-сервис	48
Создать пользовательский веб-сервис	48
Отредактировать исходный код страницы	50

С чего начать



Основы

Описание задачи

Пример. Необходимо автоматизировать работу фитнес-центра.

1. В рабочее место [*Общее*] следует добавить новый раздел [*Групповая секция*].
2. В разделе [*Групповая секция*] необходимо сформировать расписание групповых занятий.
3. Создать новую ежедневную групповую секцию можно только при наличии свободного зала. Количество залов в фитнес-центре задано системной настройкой и равно 4.
4. В каждую секцию можно добавить список групповых занятий.

Предварительные настройки

Разработка примера приложения Creatio будет выполняться на базе продукта Sales Enterprise.

Необходимо иметь облачное или локальное приложение. Самый простой вариант — получить бесплатную 14-дневную демо-версию.


Следует обеспечить доступ текущего пользователя к разделу [*Конфигурация*] ([*Configuration*]), который позволяет средствами разработки выполнять пользовательскую настройку приложения.

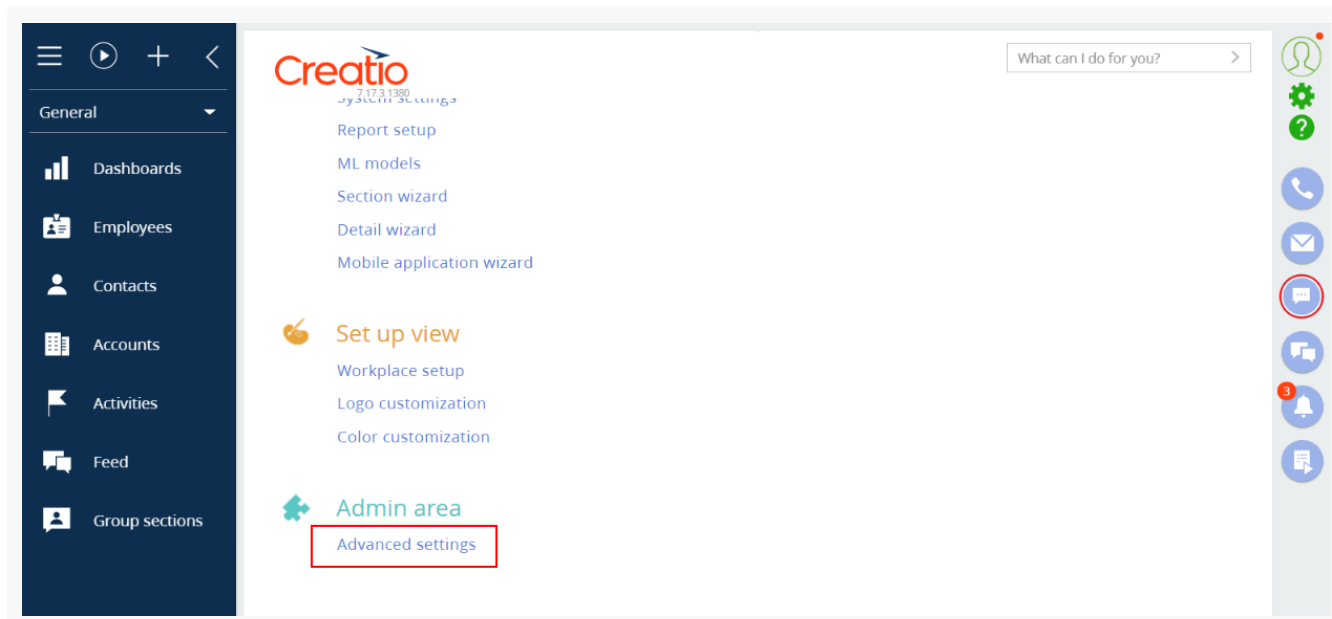
Создать пакет для разработки

[Пакет](#) — это инкапсулированный набор определенной функциональности. Его можно переносить между [рабочими средами](#) (экспортировать, устанавливать на другие приложения Creatio), предоставлять для использования другими пользователями на [Creatio Marketplace](#).

Управление пакетами осуществляется через Creatio IDE, которая реализована через раздел [*Конфигурация*] ([*Configuration*]).

Чтобы создать пакет:

1. Перейдите в раздел [*Конфигурация*] ([*Configuration*]):
 - a. Перейдите в дизайнер системы по кнопке .
 - b. В блоке [*Конфигурирование разработчиком*] ([*Admin area*]) перейдите по ссылке [*Управление конфигурацией*] ([*Advanced settings*]).



2. Чтобы создать пакет, в области работы с пакетами нажмите кнопку .

3. Заполните **свойства пакета**:

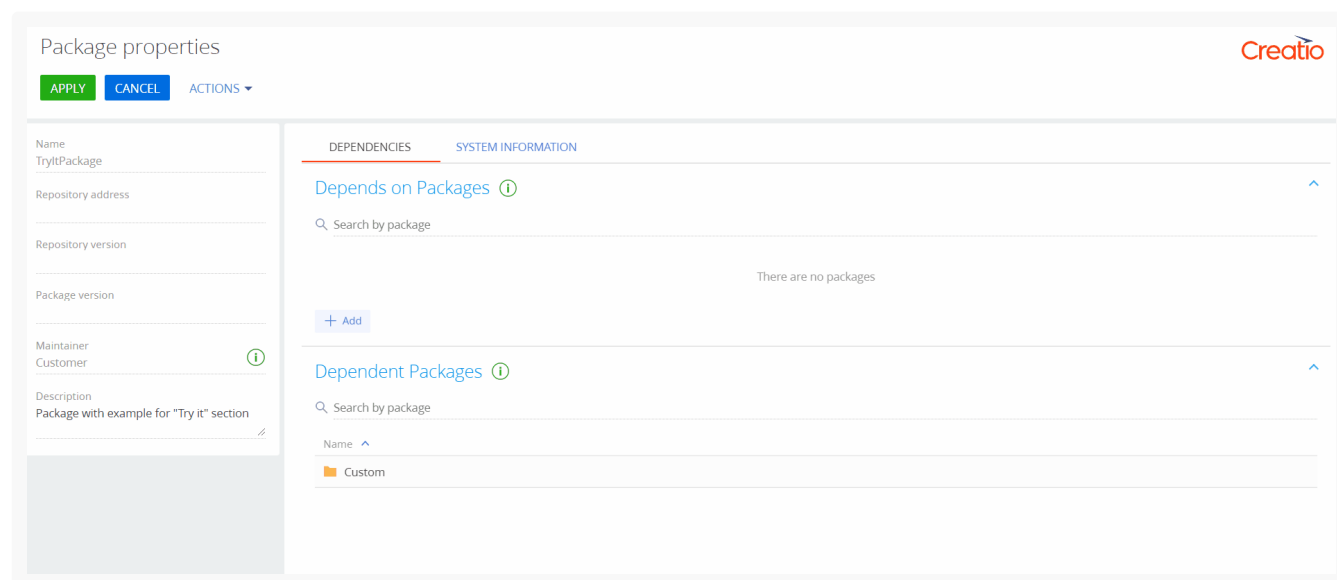
- [*Название*] ([*Name*]) — "TryItPackage";
- [*Описание*] ([*Description*]) — "Package with example for "Try it" section".

4. Добавьте зависимости пакета.

Настройка зависимостей нужна для того, чтобы иметь возможность кастомизировать существующие разделы системы, а также использовать базовые шаблоны объектов и клиентских модулей.

Чтобы добавить **зависимости пакета**:

- Нажмите кнопку [*Создать и добавить зависимости*] ([*Create and add dependencies*]).
- На вкладке [*Зависимости*] ([*Dependencies*]) на детали [*Зависит от пакетов*] ([*Depends on packages*]) установите зависимость от пакета "SalesEnterprise".




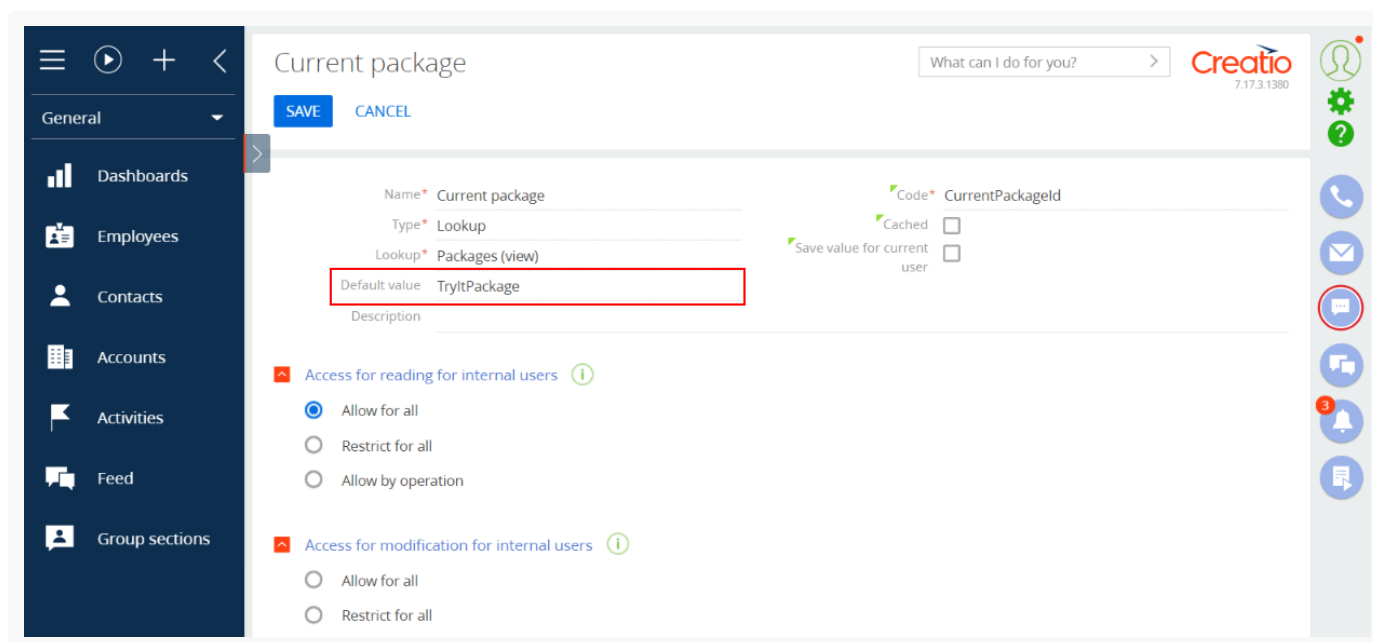
Созданный пакет будет использоваться для дальнейшей реализации примера.

Важно. Крайне не рекомендуется использовать для разработки пакет "Custom".

Переопределить текущий пакет

Перед началом разработки установите системную настройку "Текущий пакет":

1. Перейдите в дизайнер системы по кнопке .
2. В блоке [*Настройка системы*] ([*System setup*]) перейдите по ссылке [*Системные настройки*] ([*System settings*]).
3. Выберите настройку "Текущий пакет" (код `CurrentPackageId`).
4. В поле [*Значение по умолчанию*] ([*Default value*]) выберите "TryItPackage".



Сохраните изменения нажатием на [*Сохранить*] ([*Save*]).

Важно. Перед использованием **no-code инструментов** ([мастера раздела](#), мастера деталей) обязательно в системной настройке "Текущий пакет" (код `CurrentPackageId`) укажите созданный пользовательский пакет.

Итак, мы настроили пакет для разработки и установили его в системе как текущий. На [следующем шаге](#) необходимо настроить интерфейс нового раздела с помощью встроенных **no-code инструментов**.


Шаг 1. Создайте новый раздел

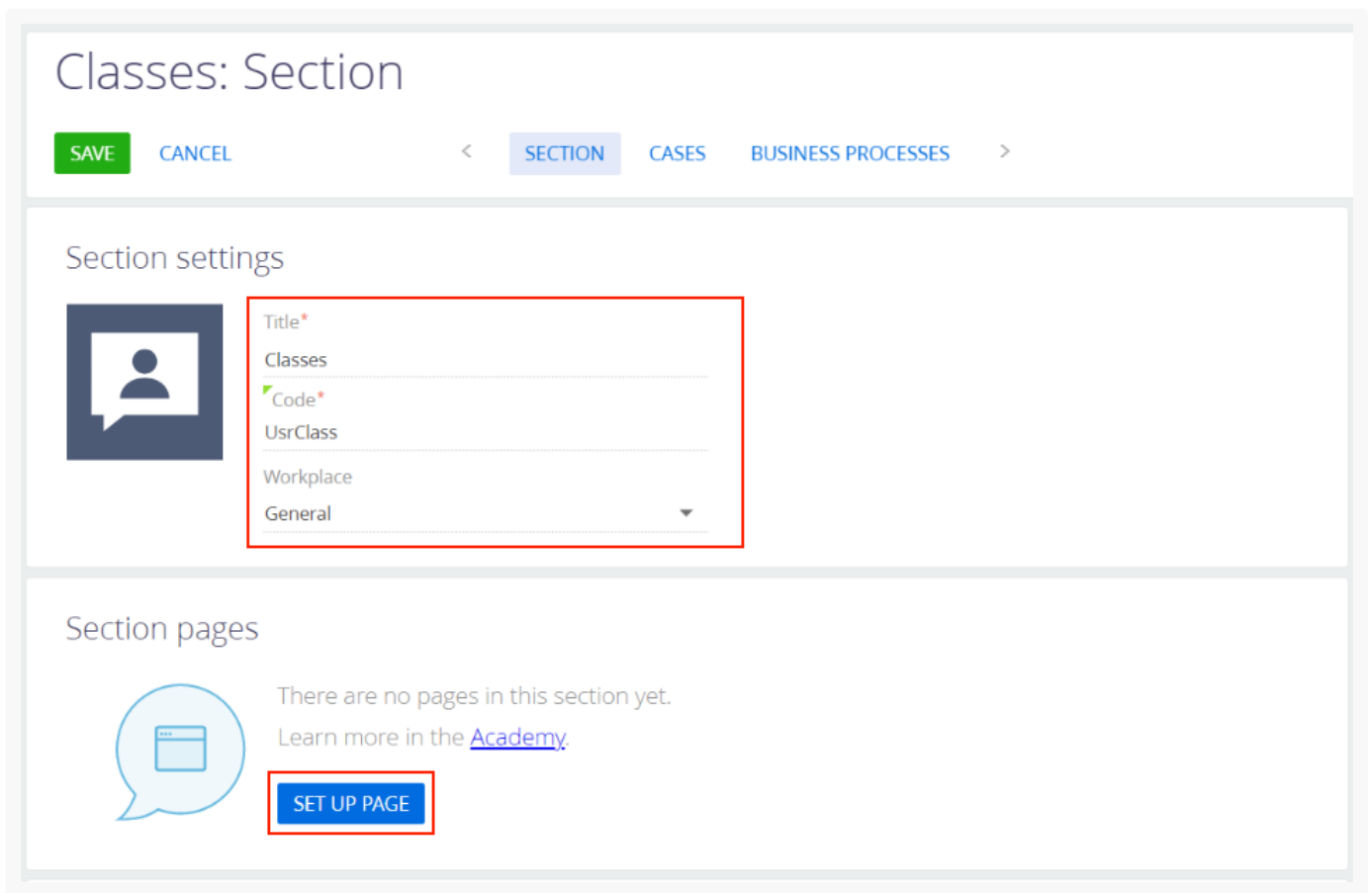
На [предыдущем шаге](#) мы настроили пакет для разработки и установили его в системе как текущий. Теперь необходимо настроить интерфейс нового раздела с помощью встроенных **no-code инструментов**.

Настроить раздел

Настройка выполняется **мастером разделов**.

Чтобы настроить раздел:

1. Перейдите в дизайнер системы по кнопке .
2. В блоке [*Настройка системы*] ([*System setup*]) перейдите по ссылке [*Мастер разделов*] ([*Section wizard*]).
3. Для нового раздела **заполните поля**:
 - [*Заголовок*] ([*Title*]) — "Групповые секции" ("Classes").
 - [*Код*] ([*Code*]) — "UsrClass".
 - [*Рабочее место*] ([*Workplace*]) — "Общее" ("General").
4. Настройте страницу раздела по кнопке [*Настроить страницу*] ([*Setup page*]).



5. **Добавьте поля** на страницу:

- а. В левом меню выберите поле с необходимым типом данных.

- b. Перетащите поле на страницу.
- c. Заполните свойства поля.

New column

SAVE CANCEL

General ^

Title *

Comment

Code *

UsrComment

Text length

Text (250 characters) v

☐ Required

☒ Copy this value when copying records

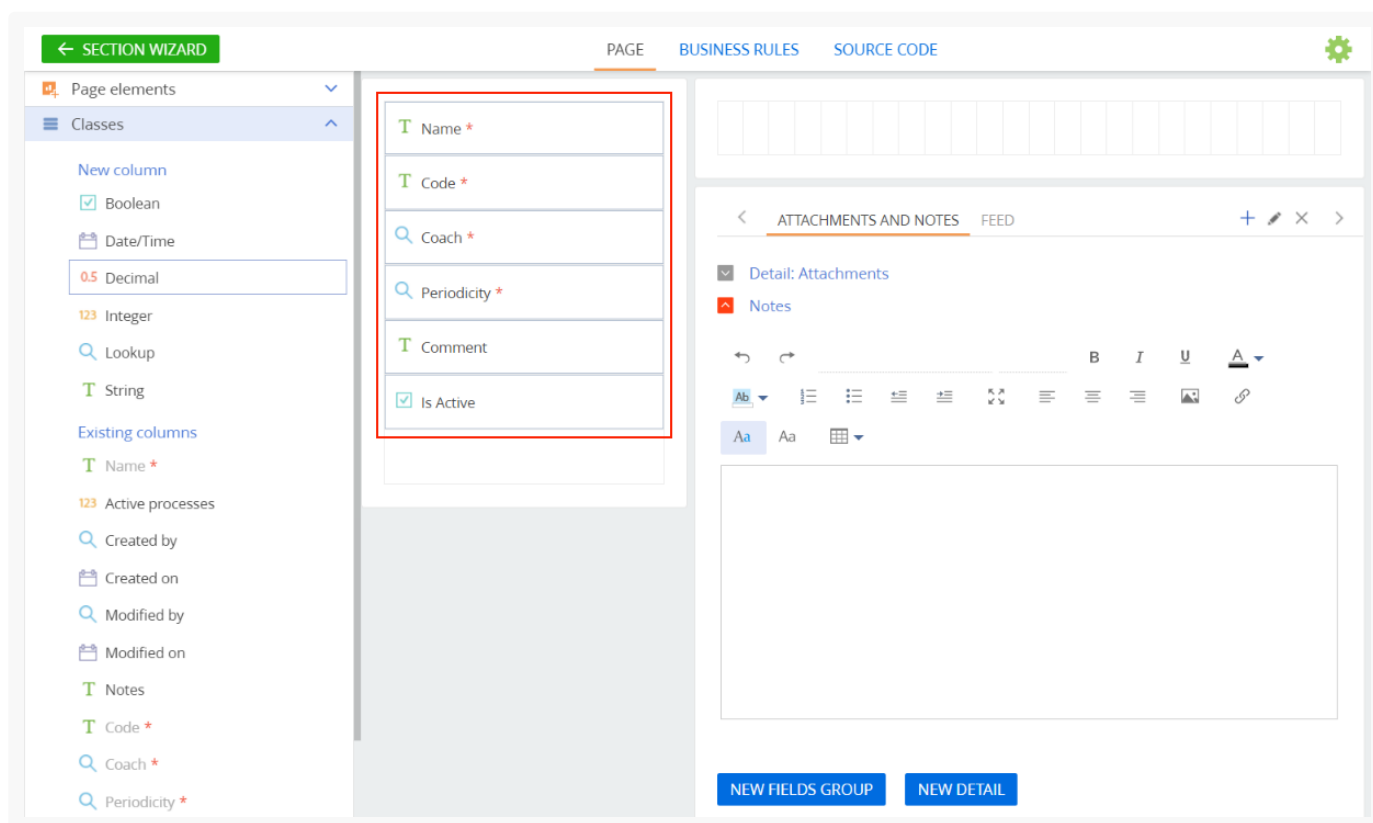
Editability ^

☐ Read-only

Appearance ^

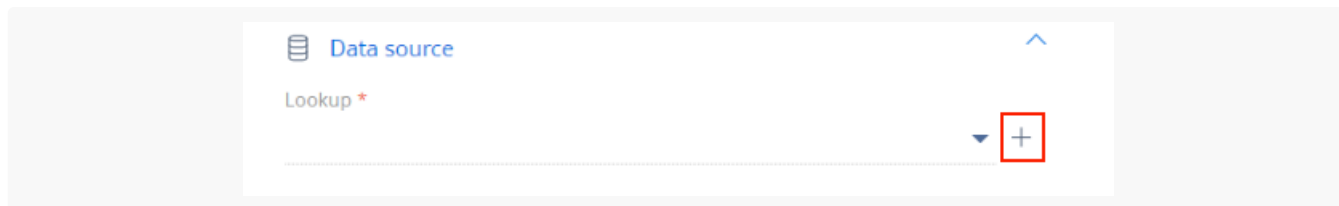
Добавьте на страницу следующие поля:

Заголовок (Title)	Код (Code)	Тип данных	Обязательность поля
Название (Name)	UserName	Строка (String)	Обязательное поле
Код секции (Code)	UserCode	Строка (String)	Обязательное поле
Тренер (Coach)	UsrCoach	Существующий справочник "Сотрудник" (Lookup "Employee")	Обязательное поле
Периодичность (Periodicity)	UsrPeriodicity	Новый справочник "Периодичность" (Lookup "Periodicity")	Обязательное поле
Комментарий (Comment)	UsrComment	Строка (String)	Необязательное поле
Активна (Is Active)	UsrIsActive	Логическое (Boolean)	Необязательное поле



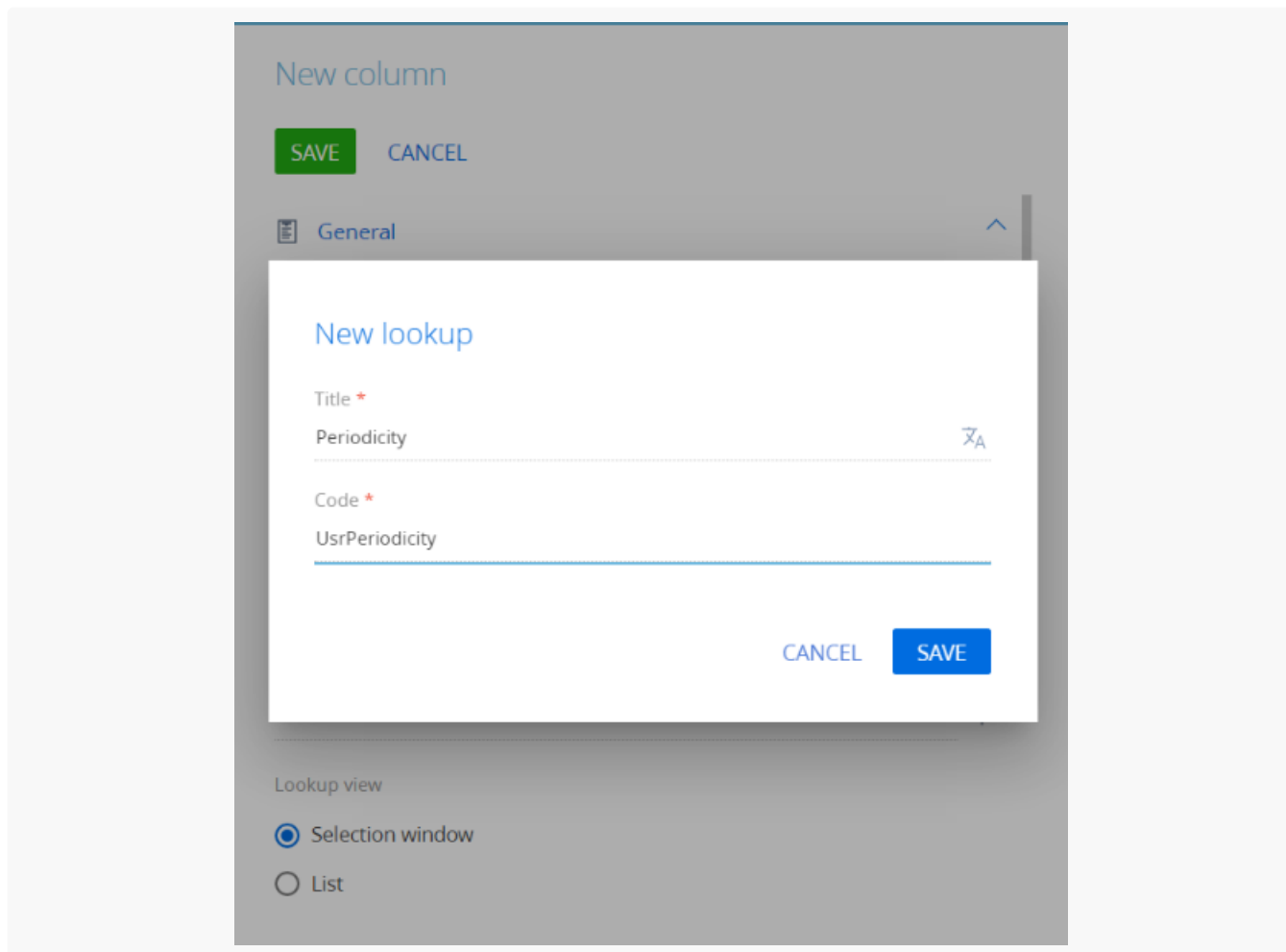
Чтобы **добавить новый справочник** для поля "Периодичность" ("Periodicity"):

- Нажмите кнопку [*Добавить новый справочник*] ([*Add new lookup*]);

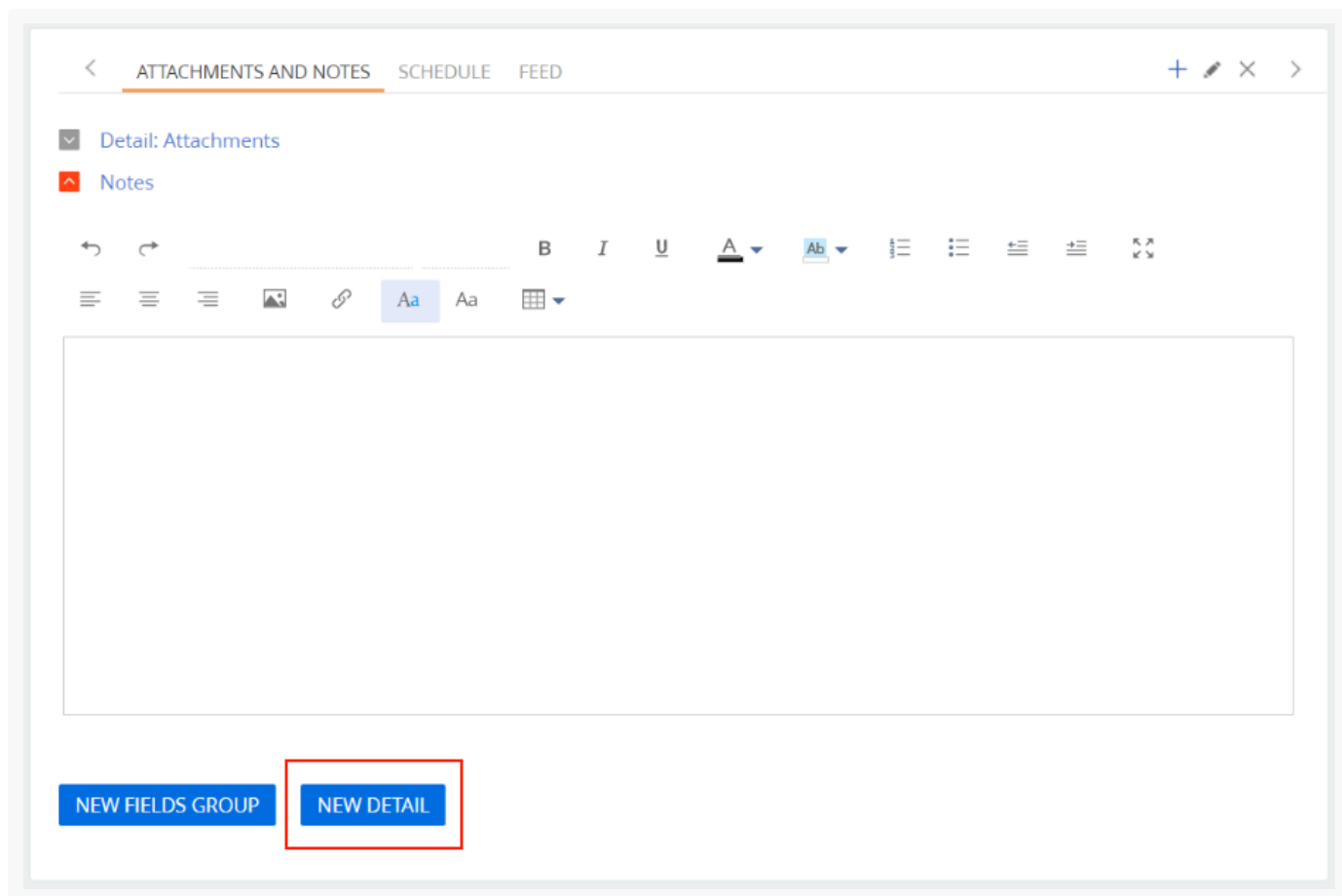


b. **Заполните поля** свойств справочника:

- [*Заголовок*] ([*Title*]) — "Периодичность" ("Periodicity");
- [*Код*] ([*Code*]) — "UsrPeriodicity".



6. Кроме полей, страница раздела будет содержать расписание связанных с секцией групповых занятий. Для отображения групповых секций на страницу раздела необходимо **добавить новую деталь** (кнопка [*Новая деталь*] ([*New detail*])).



Нажмите [*Добавить новую деталь*] —> [*Добавить по новому объекту*] ([*Add new detail*] —> [*Add using new object*]).

Detail settings

SAVE CANCEL

General

Detail *

Title *

Code *

What records to show on the page?

Where detail column *

Equals to page column *

Add using new object

Add based on existing object

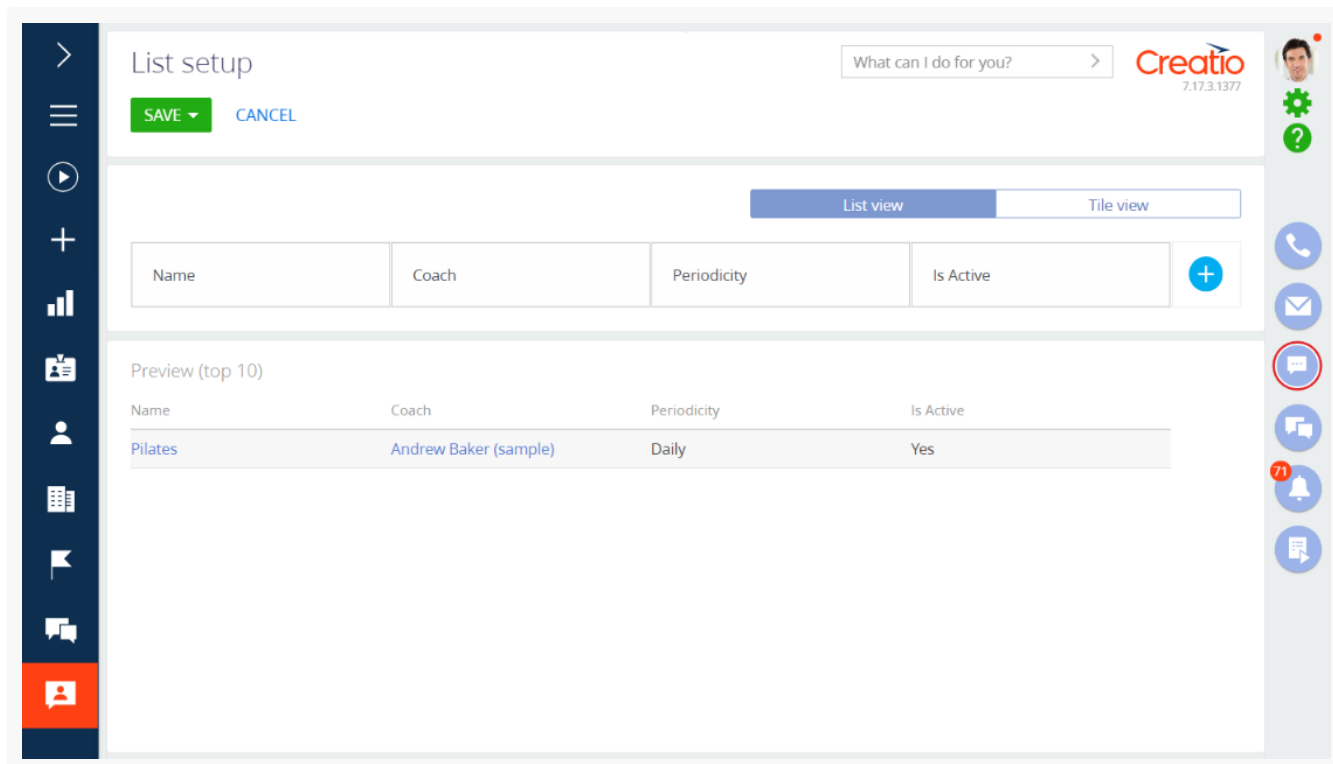
Заполните поля свойств детали:

- [*Заголовок объекта*] ([*Object title*]) — "Групповое занятие" ("Group exercise").
- [*Заголовок детали*] ([*Detail title*]) — "Групповые занятия" ("Group exercises").
- [*Код*] ([*Code*]) — "UsrGroupExercise".
- [*Сделать реестр редактируемым*] ([*Make the list editable*]) — да.
- [*Связать по справочнику*] ([*Connect by lookup*]) — "Групповые секции" ("Classes").
- [*Заголовок справочной колонки*] ([*Lookup column title*]) — "Групповая секция" ("Class").
- [*Код справочной колонки*] ([*Lookup column code*]) — "UsrClass".

7. Сохраните раздел. На этом этапе раздел уже отображается в приложении.

8. Отдельно следует **настроить отображение колонок** в реестре раздела:

- Перейдите на страницу реестра раздела.
- На панели инструментов раздела нажмите [*Вид*] —> [*Настроить колонки*] ([*View*] —> ([*Select fields to display*])).

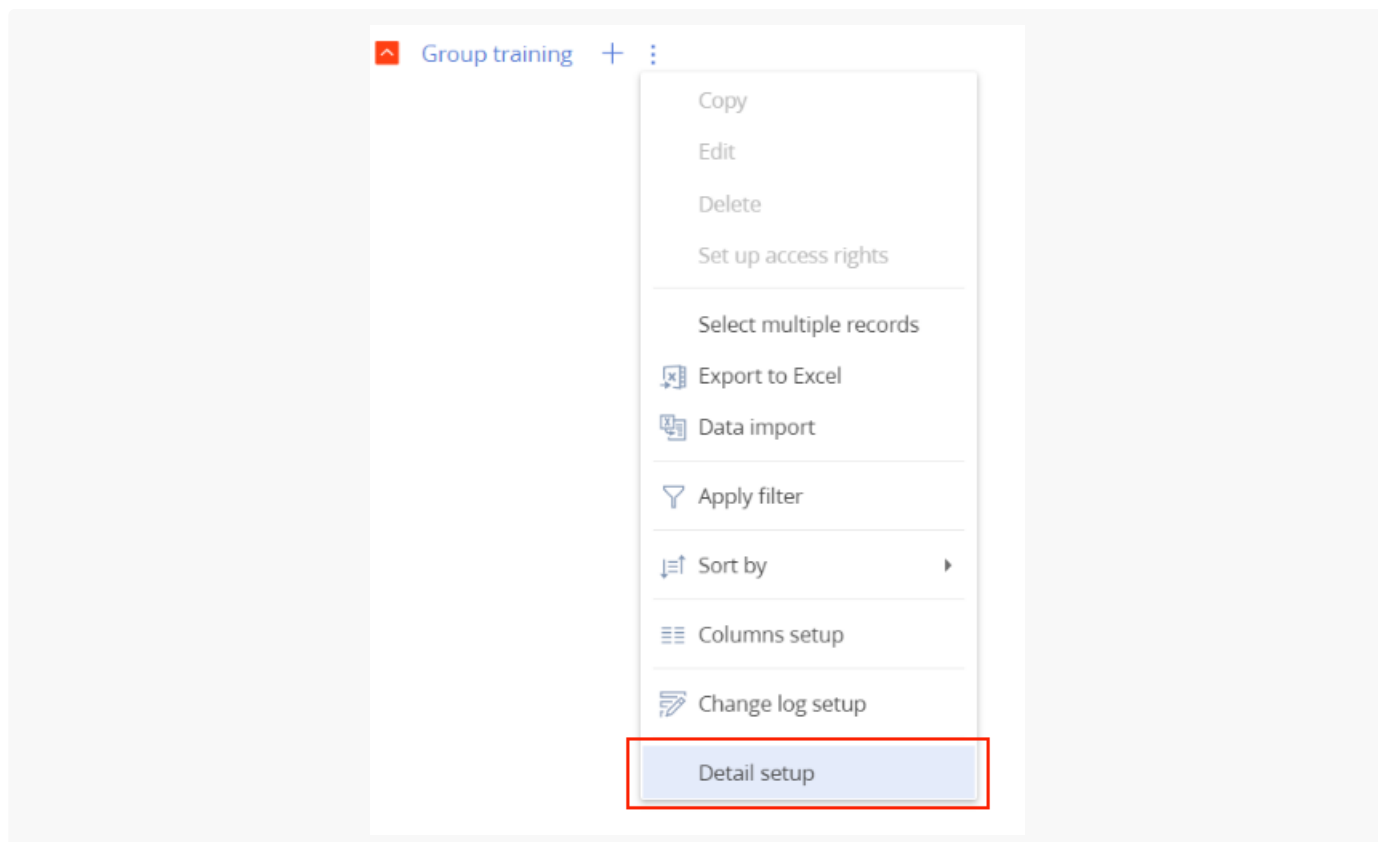


с. Для сохранения изменений нажмите [Сохранить] ([Save]).

Настроить деталь

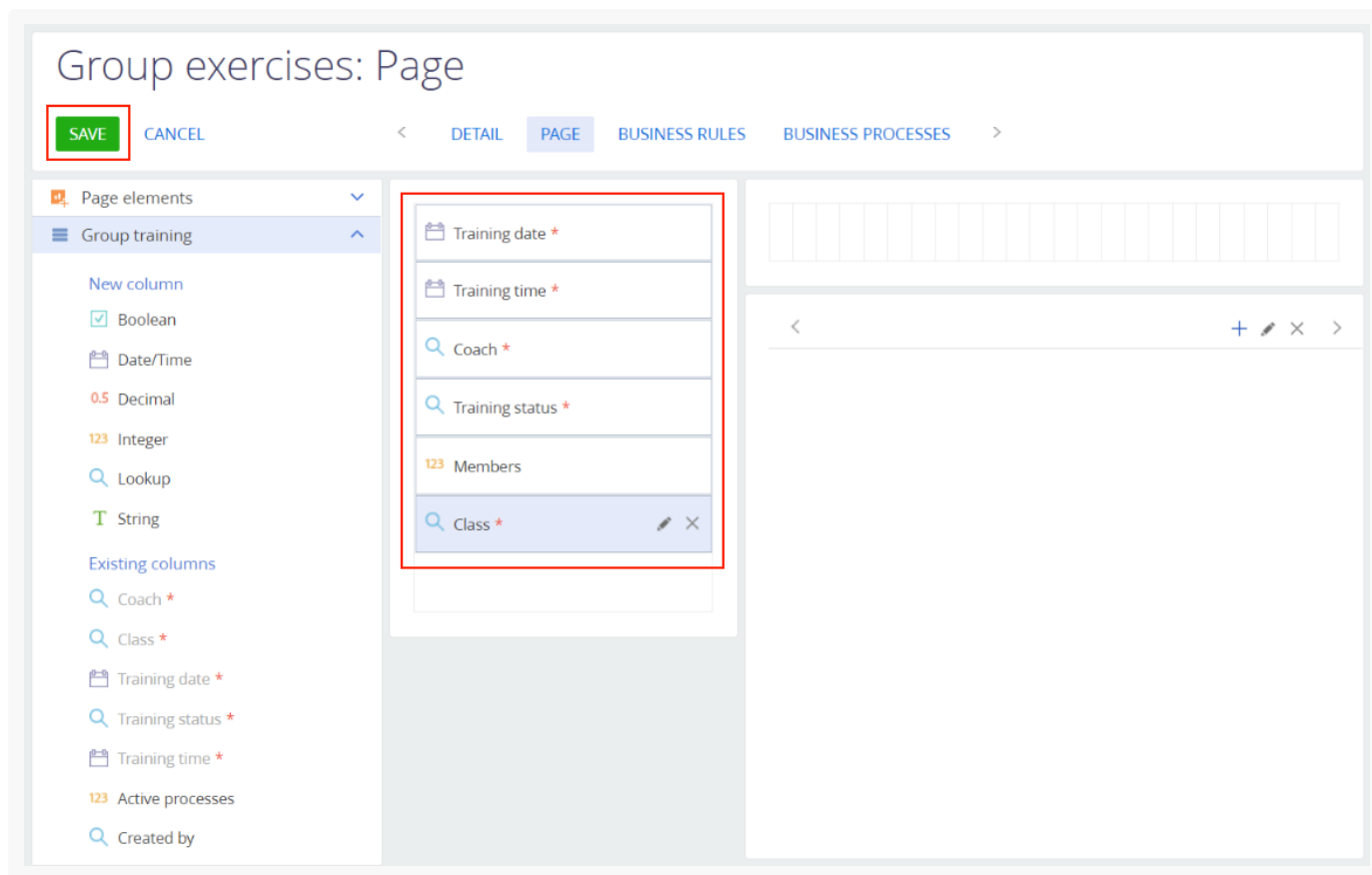
На финальном этапе создания нового раздела необходимо настроить расписание групповых занятий — добавить в деталь необходимые поля и настроить их отображение в реестре детали. Это можно сделать непосредственно на странице групповой секции.

1. Для создания новой записи в реестре раздела [Групповые секции] нажмите кнопку [Добавить] ([New]) . На открывшейся странице записи перейдите на деталь "Групповые занятия" и откройте **мастер деталей** — в меню детали выберите пункт [Настроить деталь] ([Detail setup]).



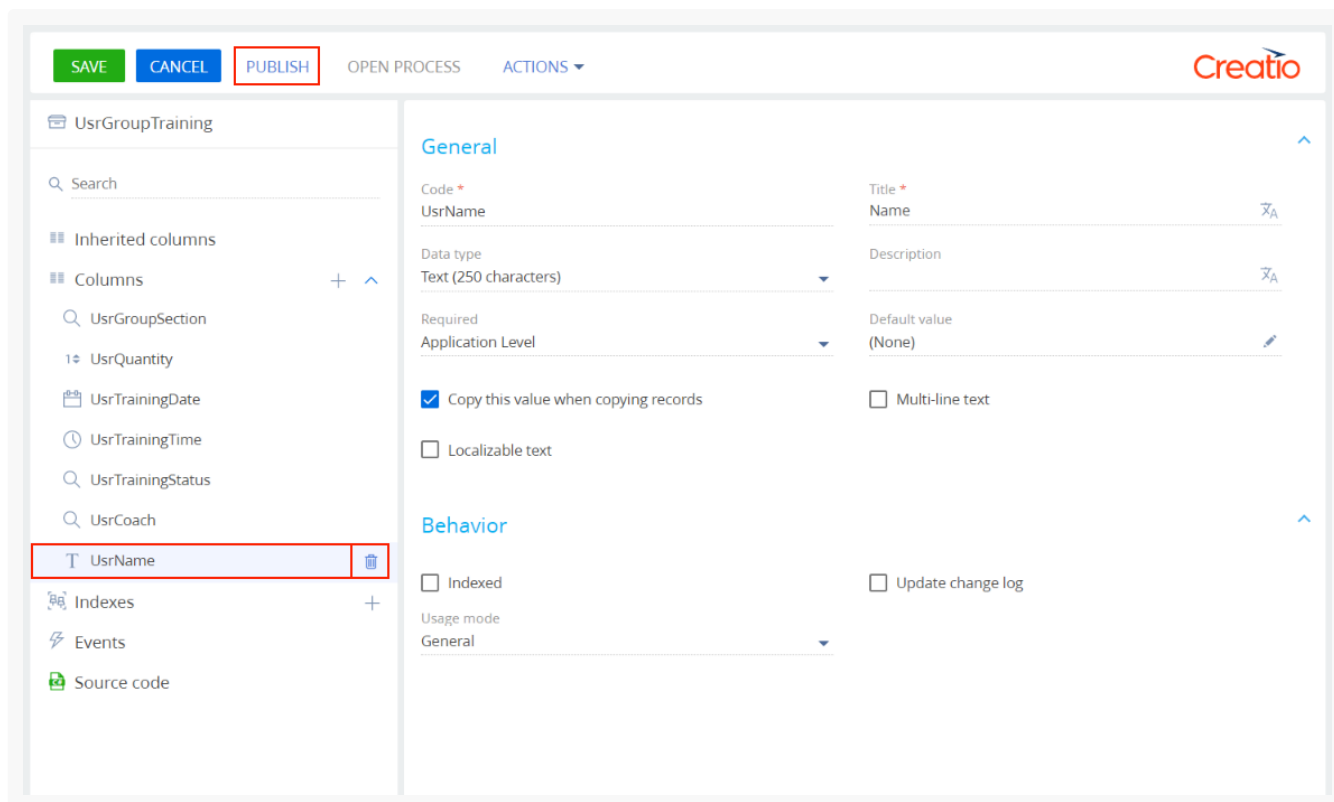
2. С помощью мастера деталей **настройте поля детали** на вкладке [*Страница*] ([*Page*]):

[Заголовок] ([Title])	[Код] ([Code])	Тип данных	Обязательность поля
Дата занятия (Training date)	UsrTrainingDate	Дата (Date)	Обязательное поле
Время занятия (Training time)	UsrTrainingTime	Время (Time)	Обязательное поле
Тренер (Coach)	UsrCoach	Существующий справочник "Сотрудник" (Lookup "Employee")	Обязательное поле
Состояние занятия (Training status)	UsrTrainingStatus	Новый справочник "Состояние занятия" (Lookup "Training status")	Обязательное поле
Количество участников (Members)	UsrQuantity	Целое число (Integer)	Необязательное поле
Групповая секция (Class)	UsrClass	Существующий справочник "Групповая секция" (Lookup "Class")	Обязательное поле



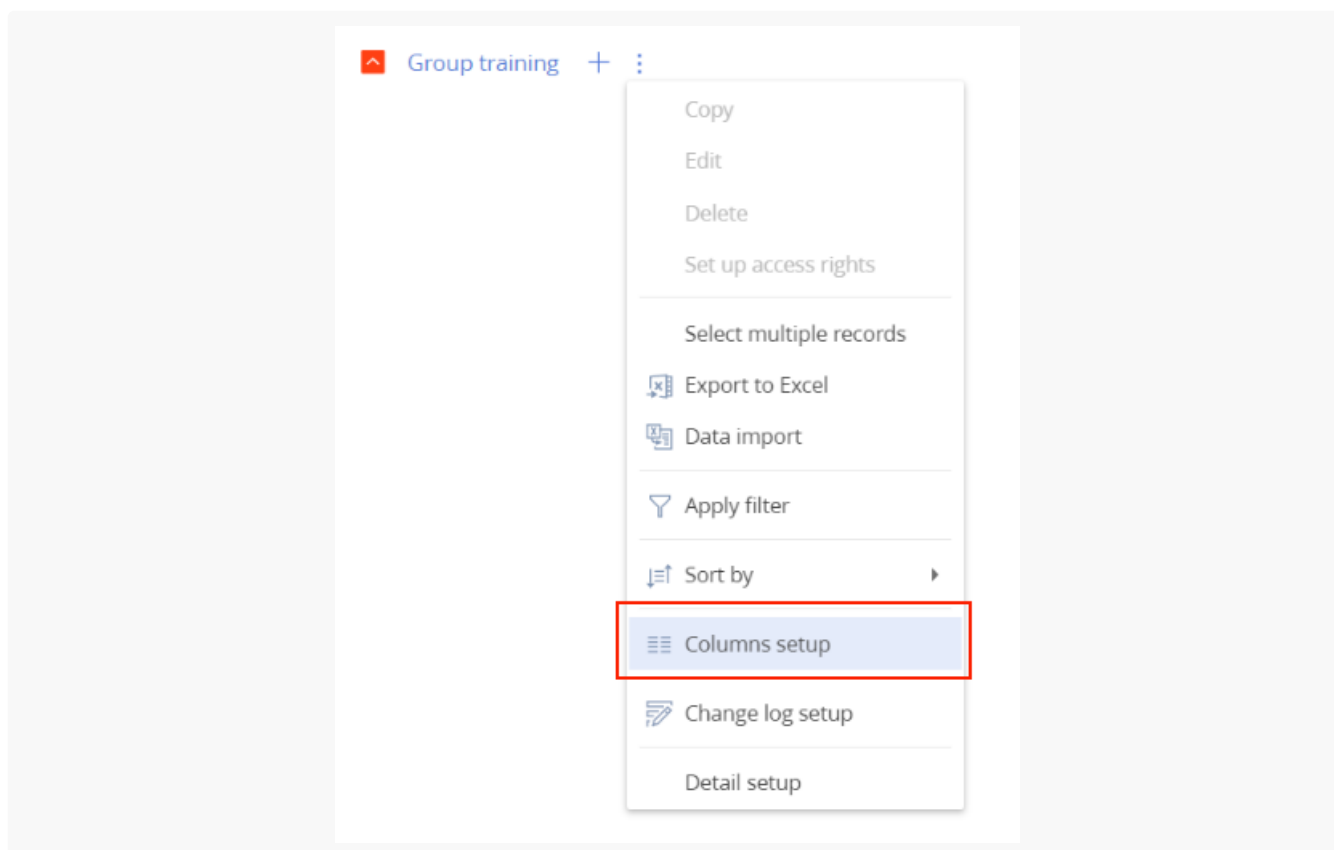
Чтобы добавить новый справочник для поля "Состояние занятия" ("Training status"):

- a. Нажмите кнопку [*Добавить новый справочник*] ([*Add new lookup*])
 - b. **Заполните поля** свойств справочника:
 - [*Заголовок*] ([*Title*]) — "Состояние занятия" ("Training status").
 - [*Код*] ([*Code*]) — "UsrTrainingStatus".
 - e. Сохраните страницу детали.
3. Обратите внимание, что при создании детали с помощью мастера разделов в нее автоматически было добавлено обязательное поле "Название" ("Name"). В нашем примере это поле не требуется и его необходимо удалить:
- a. Перейдите в раздел [*Конфигурация*] ([*Configuration*]).
 - b. В результате работы мастеров в пакете "TryItPackage" создан набор необходимых схем и прикрепленных данных. Выберите схему "UsrGroupExercise". Это схема объекта, на базе которого сформирована деталь "Групповые занятия" ("Group exercises").
 - c. Двойным кликом мыши откройте схему.
 - d. Удалите колонку "UsrName" из списка колонок объекта.
 - e. Опубликуйте схему, нажав на кнопку [*Опубликовать*] ([*Publish*]).

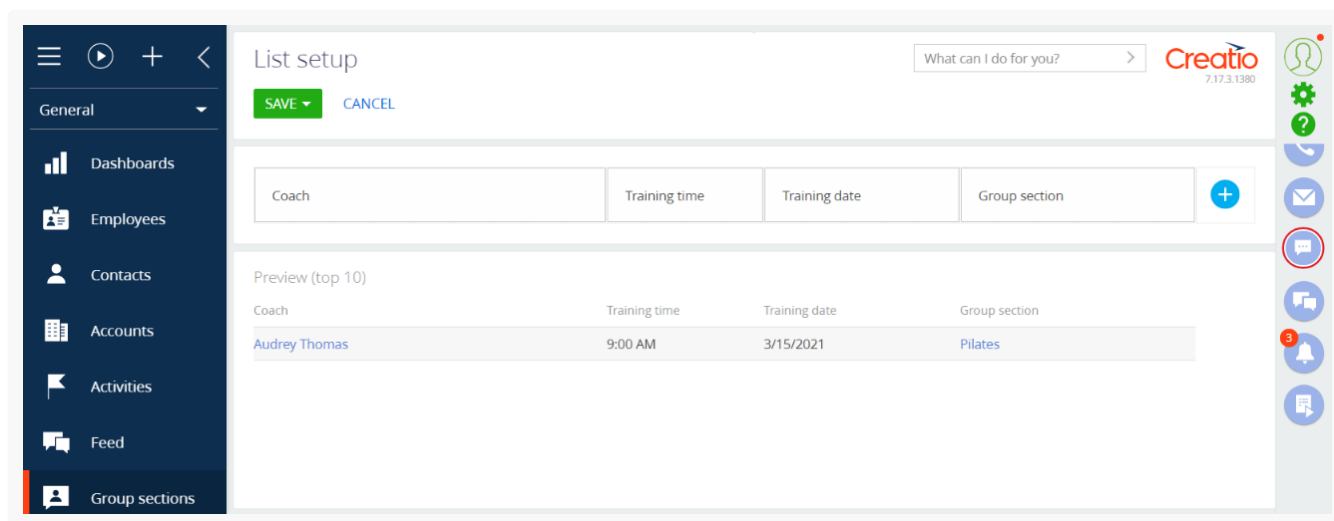


4. Деталь с расписанием почти готова, осталось **настроить колонки детали**:

а. В меню детали выберите пункт [*Настройка колонок*] ([*Columns setup*]).



б. Настройте колонки детали.



с. Для сохранения изменений нажмите [Сохранить] ([Save]).

В результате мы получили полностью настроенный интерфейс нового раздела. На [следующем шаге](#) необходимо:

- заполнить данными новые справочники;
- добавить системную настройку, которая определяет количество залов;
- прикрепить все новые данные к пакету разработки.

Шаг 2. Добавьте новые данные

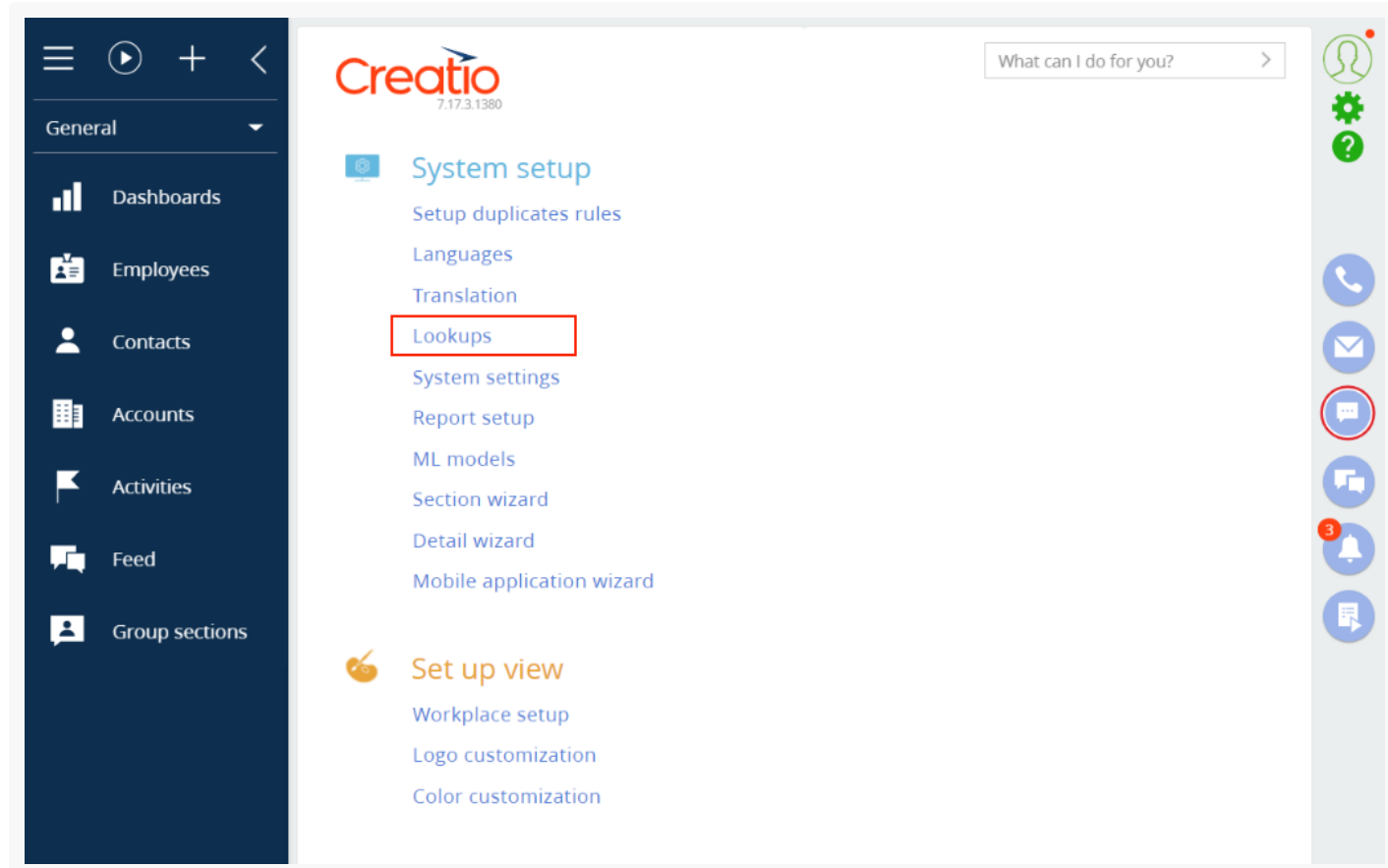
Основы

На [предыдущем шаге](#) мы получили полностью настроенный интерфейс нового раздела. Теперь необходимо:


- заполнить данными новые справочники;
- добавить системную настройку, которая определяет количество залов;
- прикрепить все новые данные к пакету разработки.

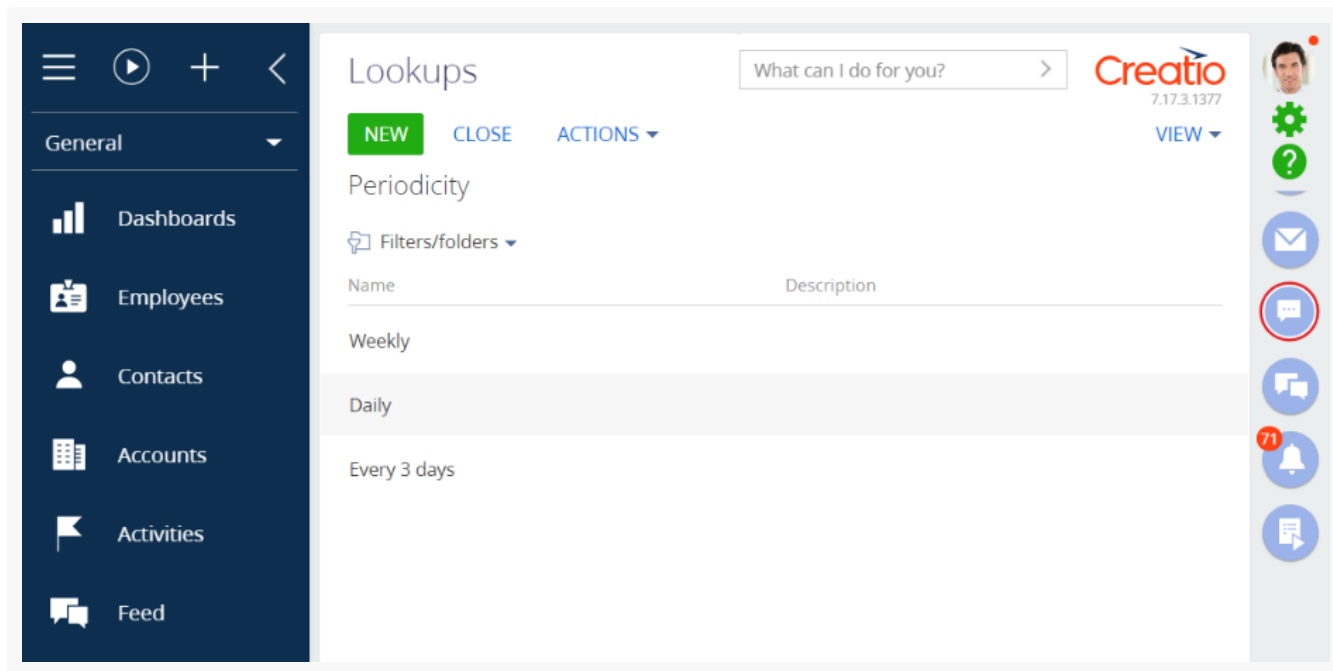
Заполнить справочники

Перечень возможных значений для заполнения полей настраивается в разделе [Справочники] ([Lookups]) дизайнера системы.



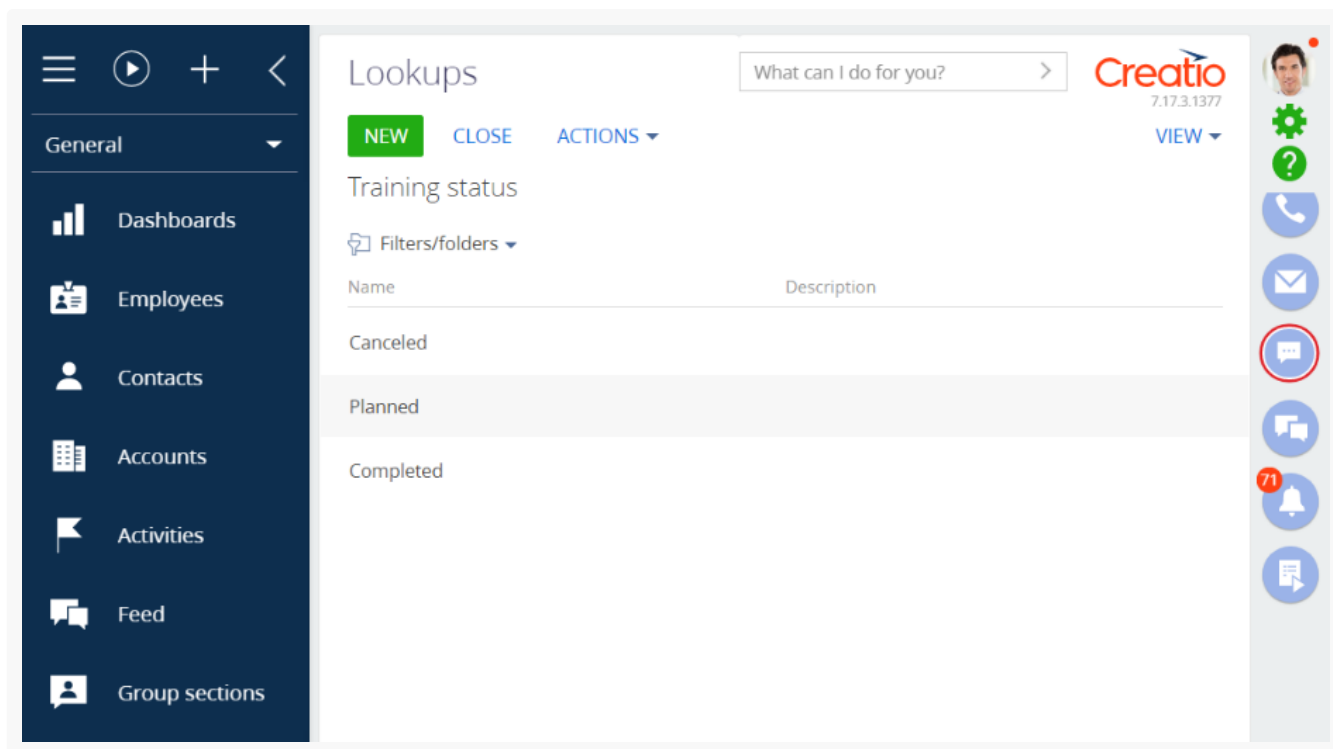
1. Заполните справочник "Периодичность" ("Periodicity"):

- a. Перейдите в дизайнер системы по кнопке .
- b. В блоке [*Настройка системы*] ([*System setup*]) перейдите по ссылке [*Справочники*] ([*Lookups*]).
- c. Используя фильтр в верхней части страницы, найдите справочник "Периодичность" ("Periodicity").
- d. Откройте наполнение справочника и заполните его значениями:
 - Ежедневно (Daily);
 - Через 3 дня (Every 3 days);
 - Еженедельно (Weekly).



2. Заполните справочник "Состояние занятия" ("Training status"):


- a. Перейдите в раздел [Справочники] ([Lookups]).
- b. Используя фильтр в верхней части страницы, найдите справочник "Состояние занятия" ("Training status").
- c. Откройте наполнение справочника и заполните его значениями:
 - Запланировано (Planned);
 - Отменено (Canceled);
 - Завершено (Completed).

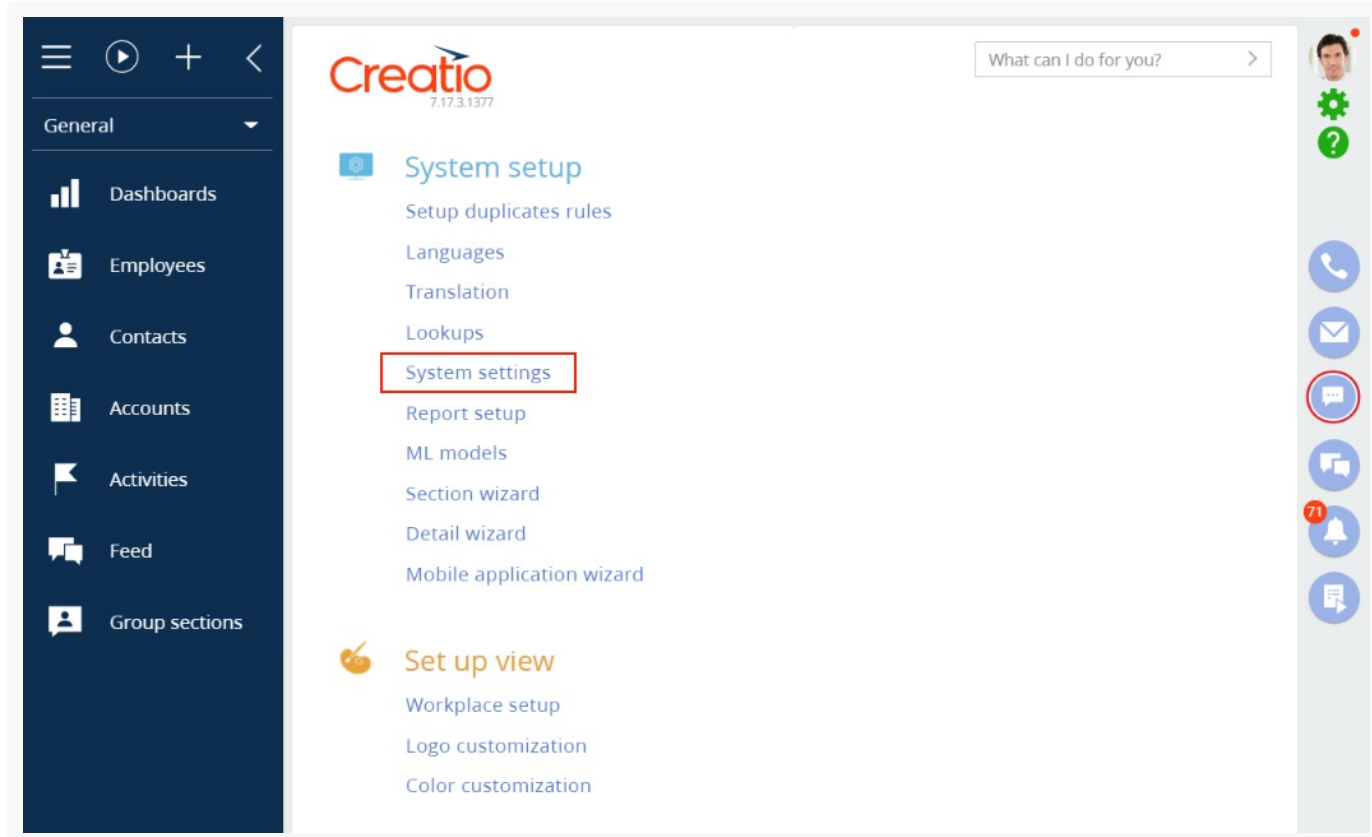


Создать системную настройку

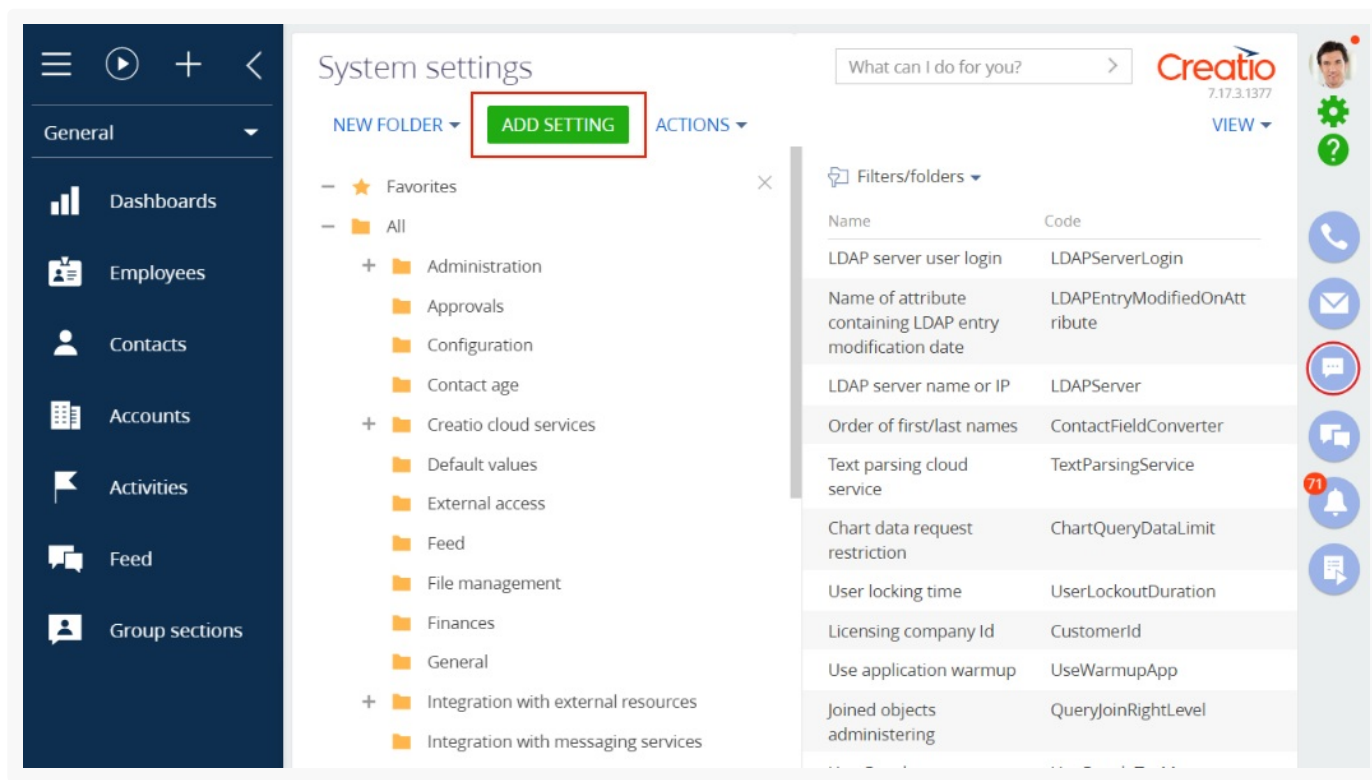
Для нашего примера необходимо зафиксировать в системе количество залов для групповых занятий. Для этого следует использовать системную настройку.

Все базовые и пользовательские системные настройки находятся в разделе [Системные настройки] ([*System settings*])

1. Перейдите в дизайнер системы по кнопке .
2. В блоке [Настройка системы] ([*System setup*]) перейдите по ссылке [Системные настройки] ([*System settings*]).



3. Чтобы добавить системную настройку, нажмите [*Добавить настройку*] ([*Add settings*]).



4. Заполните поля **свойств** системной настройки:

- [*Название*] ([*Name*]) — "Количество залов для групповых секций" ("Number of gyms").

- [Код] ([Code]) — "GymsNumber".
- [Тип] ([Type]) — "Целое число" ("Integer").
- [Значение по умолчанию] ([Default value]) — 4.

The screenshot shows the 'Number of gyms' configuration page in the Creatio system. The left sidebar contains navigation links: Dashboards, Employees, Contacts, Accounts, Activities, Feed, and Group sections. The main area is titled 'Number of gyms' and includes a search bar and 'SAVE' and 'CANCEL' buttons. The configuration fields are as follows:

Field	Value
Name*	Number of gyms
Type*	Integer
Default value	4
Description	
Code*	GymsNumber
Cached	<input checked="" type="checkbox"/>
Save value for current user	<input type="checkbox"/>

Below the fields, there are two sections for access control:

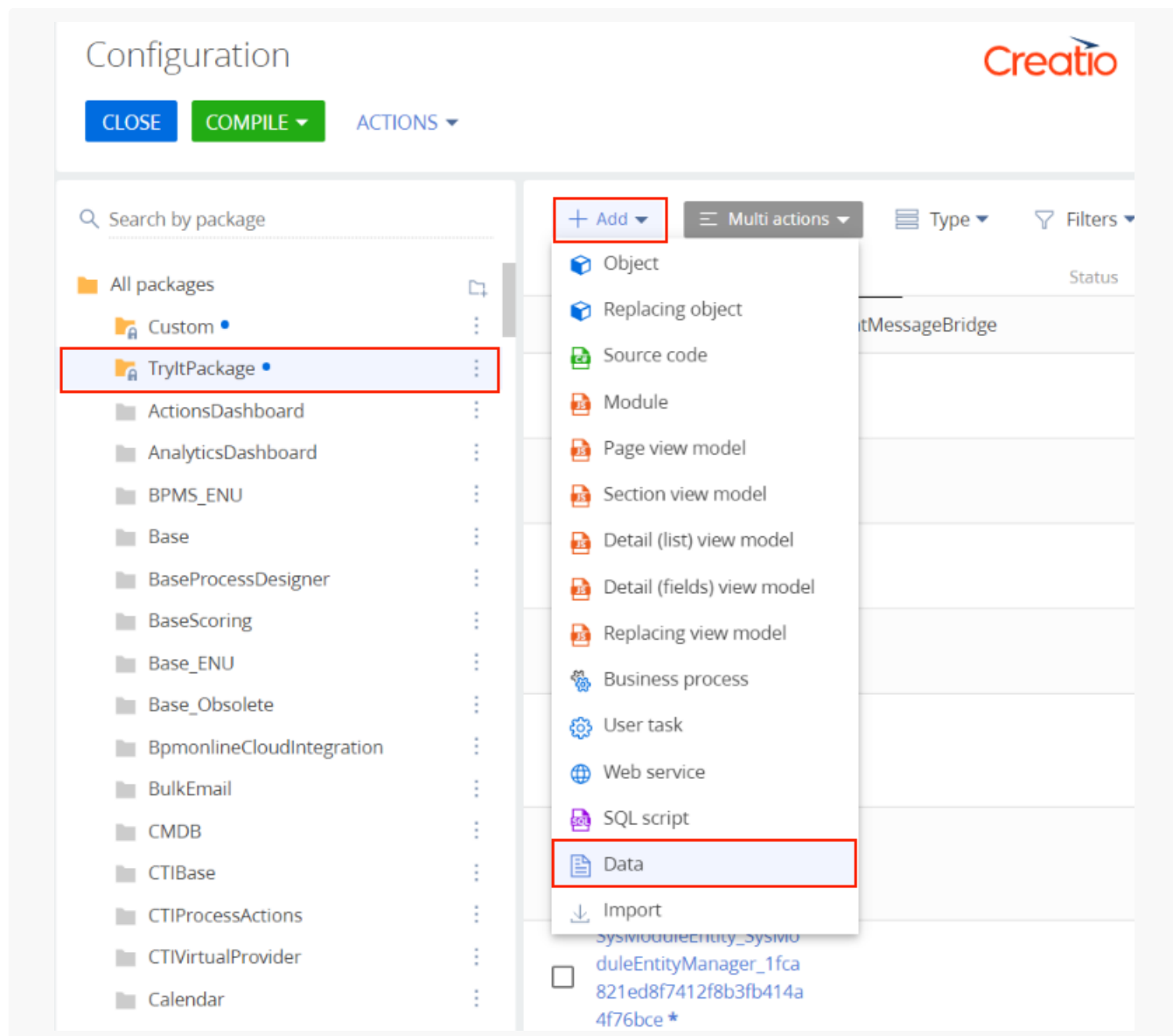
- Access for reading for internal users** (with an information icon):
 - ☒ Allow for all
 - ☐ Restrict for all
 - ☐ Allow by operation
- Access for modification for internal users** (with an information icon):
 - ☒ Allow for all
 - ☐ Restrict for all
 - ☐ Allow by operation

The right sidebar contains various system icons, including a user profile, settings, help, and communication tools.

Привязать данные к пакету

Данные новых справочников и системной настройки хранятся в базе данных приложения для среды разработки. Чтобы реализовать возможность перенести разработанную функциональность в предпромышленную и промышленную среду, необходимо прикрепить к пакету разработки новые данные. Для этого в Creatio используется механизм **привязки данных к пакету**.

1. Добавьте данные справочника "Периодичность" ("Periodicity") к пакету:
 - a. Перейдите в раздел [Конфигурация] ([Configuration]) и выберите пакет "TryItPackage".
 - b. На панели инструментов рабочей области нажмите кнопку [Добавить] ([Add]) и выберите в списке вид конфигурационного элемента [Данные] ([Data]).



- с. На странице в поле [Объект] ([Object]) выберите "UsrPeriodicity" — объект справочника "Периодичность" ("Periodicity").

SAVE
CANCEL
Creatio

Name *

Object *

UsrPeriodicity ▼

Installation type *

Installation ▼

Package

TryItPackage

COLUMNS SETTING BOUND DATA

Select an object, then set up the columns

+ Add

i Note

Installation type

The installation type determines the method of adding the application data upon the installation of the package.

- Installation - the data is added on the first installation of the package and is updated along with the package updates. This is the most versatile type and is used by default.

- d. Проверьте, что на вкладке [*Настройка колонок*] ([*Columns Setting*]) выбраны все колонки, которые содержат данные объекта.

CLOSE
Creatio

Name

UsrPeriodicity

Object

UsrPeriodicity ▼

Installation type

Installation ▼

Package

TryItPackage

COLUMNS SETTING BOUND DATA

Caption ^	Name	Forced update	Key
Id	Id	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Name	Name	<input type="checkbox"/>	<input type="checkbox"/>

+ Add

i Note

Installation type

The installation type determines the method of adding the application data upon the installation of the package.

- Installation - the data is added on the first installation of the package and is updated along with the package updates. This is the most versatile type and is used by default.

- e. Перейдите на вкладку [*Привязанные данные*] ([*Bound data*]) и нажмите кнопку [*Добавить*] ([*Add*]).

SAVE CANCEL

Creatio

Name *
UsrPeriodicity

Object *
UsrPeriodicity

Installation type *
Installation

Package
TryItPackage

COLUMNS SETTING BOUND DATA

Search

+ Add

Select records to bind

Note

Installation type

The installation type determines the method of adding the application data upon the installation of the package.

- Installation - the data is added on the first installation of the package and is updated along with the package updates. This is the most versatile type and is used by default.
- Update existing - the data is added during the first installation of the package. Updating the package modifies only the data in the columns where the "Error update" checkbox is

f. В открывшемся окне выберите все записи.

Select: Periodicity

Name Search

Selected: 3

Id	Name
65c94e6b-0b9f-42c9-9700-b161e16ef171	Daily
f488ba25-3824-4b2a-a822-604b86265b74	Every 3 days
76c1eb09-964b-4096-ae8e-863a675ba425	Weekly

CANCEL SELECT

g. Сохраните элемент.

2. Аналогичным образом добавьте данные справочника "Состояние занятия" ("Training status"). В качестве объекта укажите "UsrTrainingStatus".

3. Привязка к пакету системной настройки подразумевает выполнение привязки кода системной настройки и значения системной настройки.

- a. Добавьте к пакету новый элемент типа [Данные] ([Data]).
- b. В поле [Объект] ([Object]) укажите "SysSettings" — таблица данных, хранящая коды всех системных настроек приложения.

SAVE CANCEL ACTUALIZE DATA

Creatio

Name *
SysSettings

Object
SysSettings

Installation type *
Installation

Package
TryItPackage

Note

Installation type

The installation type determines the method of adding the application data upon the installation of the package.

- Installation - the data is added on the first installation of the package and is updated along with the package updates. This is the most versatile type and is used by default.
- Update existing - the data is added during the first installation of the package. Updating the package modifies only the data in the columns where the "Forced update" checkbox is

COLUMNS SETTING BOUND DATA

<input type="checkbox"/> Caption ^	Name	Forced update	Key
<input type="checkbox"/> Id	Id	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> Name	Name	<input type="checkbox"/>	<input type="checkbox"/>

+ Add

- c. На вкладке [Привязанные данные] ([Bound data]) нажмите кнопку [Добавить] ([Add]) и, в открывшемся окне, с помощью фильтра найдите настройку "Количество залов для групповых секций" ("Number of gyms").

Select: System setting
×

Name
▼
×

Selected: 1

<input checked="" type="checkbox"/> Id	Name ^
<input checked="" type="checkbox"/> d2dcff68-5222-489b-a607-4fb7290b6aee	Number of gyms

CANCEL
SELECT

- d. Сохраните элемент.
- e. Создайте еще один элемент типа [Данные] ([Data]), где в поле [Объект] ([Object]) укажите "SysSettingsValue" — таблица данных, хранящая значения всех системных настроек приложения.
- f. На вкладке [Настройки колонок] ([Column settings]) добавьте колонки [Целое число] ([Integer]) и [Системная настройка] ([System setting]).

SAVE CANCEL Creatio

Name *
SysSettingsValue

Object *
SysSettingsValue

Installation type *
Installation

Package
TryItPackage

Note
Installation type
The installation type determines the method of adding the application data upon the installation of the package.
• Installation - the data is added on the first installation of the package and is updated along with the package updates. This is the most versatile type and is used by default.
• Update existing - the data is added during the first installation of the package. Updating the package modifies only the data in the columns where the "Forced update" checkbox is

COLUMNS SETTING BOUND DATA

	Name	Forced update	Key
<input type="checkbox"/> Caption			
<input checked="" type="checkbox"/> Id	Id	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Integer	IntegerValue	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> System setting	SysSettings	<input type="checkbox"/>	<input type="checkbox"/>

+ Add

- g. На вкладке [Привязанные данные] ([Bound data]) с помощью фильтра по названию выберите данные для настройки "Количество залов для групповых секций" ("Number of gyms").

Select: System setting value

System setting Number of gyms

Selected: 1

Id	Integer	System setting
<input checked="" type="checkbox"/> 70dca3b7-d205-4220-b471-aea4e48d8aa4	4	Number of gyms

CANCEL SELECT

- h. Сохраните элемент.

В результате мы добавили к интерфейсу необходимые для работы примера данные и прикрепили их к

пакету разработки. На [следующем шаге](#) необходимо реализовать бизнес-логику примера.

Шаг 3. Добавьте валидацию страницы

Оснoвы

На [предыдущем шаге](#) мы добавили к интерфейсу необходимые для работы примера данные и прикрепили их к пакету разработки.

Теперь реализуем бизнес-логику примера.

Условие примера.

- Ежедневная групповая секция добавляется в расписание только при наличии свободного зала.
- Количество залов в фитнес-центре задано системной настройкой и равно 4.
- При добавлении или редактировании ежедневной групповой секции необходимо выполнять проверку: если в результате изменений общее число активных ежедневных секций в системе будет превышать значение системной настройки, то сохранение запрещается и выводится сообщение "Нет свободных залов, допускается не более N секций", где N — значение системной настройки.

Для этого необходимо изменить исходный код клиентского модуля, который отвечает за работу страницы раздела.

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 только схемы с типом [Клиентский модуль] ([Client module]).

The screenshot shows the 'Configuration' window in Creatio. On the left, the 'TryItPackage' is selected in the package list. On the right, the 'Client module' filter is applied to the schema list. The table below represents the data shown in the screenshot:

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.js

```

define("UsrGroupSections1Page", [], function() {
    return {
        entitySchemaName: "UsrGroupSections",
        messages: {},
        attributes: {
            /* Атрибут, который хранит текущее количество активных ежедневных секций. */
            "responseCollectionTrainings": {
                "dataValueType": Terrasoft.DataValueType.INTEGER,
                "type": Terrasoft.ViewModelColumnType.VIRTUAL_COLUMN
            },
            /* Атрибут, который хранит значение системной настройки. */
            "maximumDailyActiveSections": {
                "dataValueType": Terrasoft.DataValueType.INTEGER,
                "type": Terrasoft.ViewModelColumnType.VIRTUAL_COLUMN
            }
        },
        methods: {
            /* Запускается при загрузке схемы страницы и вызывает метод подсчета текущего кол
            onEntityInitialized: function(){
                this.callParent(arguments);
                this.getPeriodicityActiveNumber();
                this.getMaximumDailyActiveSections();
            },
            /* Вычисляет текущее количество активных ежедневных секций и записывает полученно
            getPeriodicityActiveNumber: function() {
                var periodicity = "Ежедневно";
                var esqPeriodicity = this.Ext.create("Terrasoft.EntitySchemaQuery", {
                    rootSchemaName: "UsrGroupSections"
                });
                esqPeriodicity.addColumn("UsrName");
                var groupFilters = this.Ext.create("Terrasoft.FilterGroup");
                var filterPerodicity = this.Terrasoft.createColumnFilterWithParameter(this.Te
                var thisId = this.get("Id");
                var filterId = this.Terrasoft.createColumnFilterWithParameter(this.Terrasoft.
                var filterIsActive = this.Terrasoft.createColumnFilterWithParameter(this.Terr
                groupFilters.addItem(filterPerodicity);
                groupFilters.logicalOperation = this.Terrasoft.LogicalOperatorType.AND;
                groupFilters.addItem(filterIsActive);
                groupFilters.logicalOperation = this.Terrasoft.LogicalOperatorType.AND;
                groupFilters.addItem(filterId);
                esqPeriodicity.filters.add(groupFilters);
                esqPeriodicity.getEntityCollection(function(result) {
                    if (!result.success) {
                        this.showInformationDialog("Request error");
                        return;
                    }
                }
            }
        }
    };
});

```

```

        else {
            var lengthCollection = result.collection.collection.length;
            this.set("responseCollectionTrainings", lengthCollection);
        }
    }, this);
},
/* Добавляет валидацию к полю "Периодичность". При изменении данного поля либо со
setValidationConfig: function() {
    this.callParent(arguments);
    this.addColumnValidator("UsrPeriodicity", this.periodicityValidator);
},
/* Метод-валидатор – если секция ежедневная, сравнивает текущее количество активн
periodicityValidator: function() {
    var invalidMessage= "";
    var periodicity = this.get("UsrPeriodicity").displayValue;
    if (periodicity=="Daily") {
        var isActive = this.get("UsrIsActive");
        var myVariable = this.get("maximumDailyActiveSections");
        var lengthCollection = this.get("responseCollectionTrainings");
        if (lengthCollection >= myVariable && isActive) {
            invalidMessage = "The number of gyms is limited. No more than " + myV
        }
    }
    else {
        invalidMessage = "";
    }
    return {
        invalidMessage: invalidMessage
    };
},
/* Получает значение системной настройки "GymsNumber". */
getMaximumDailyActiveSections: function() {
    var myVariable;
    var callback = function(value) {
        myVariable = value;
    };
    this.Terrasoft.SysSettings.querySysSettingsItem("GymsNumber", callback, this)
    if (myVariable === undefined) {
        return;
    }
    else {
        this.set("maximumDailyActiveSections", myVariable);
    }
}
},
modules: /**SCHEMA_MODULES*/{/**SCHEMA_MODULES*/,
/* Без изменений. */
details: /**SCHEMA_DETAILS*/{

```



```

// ...

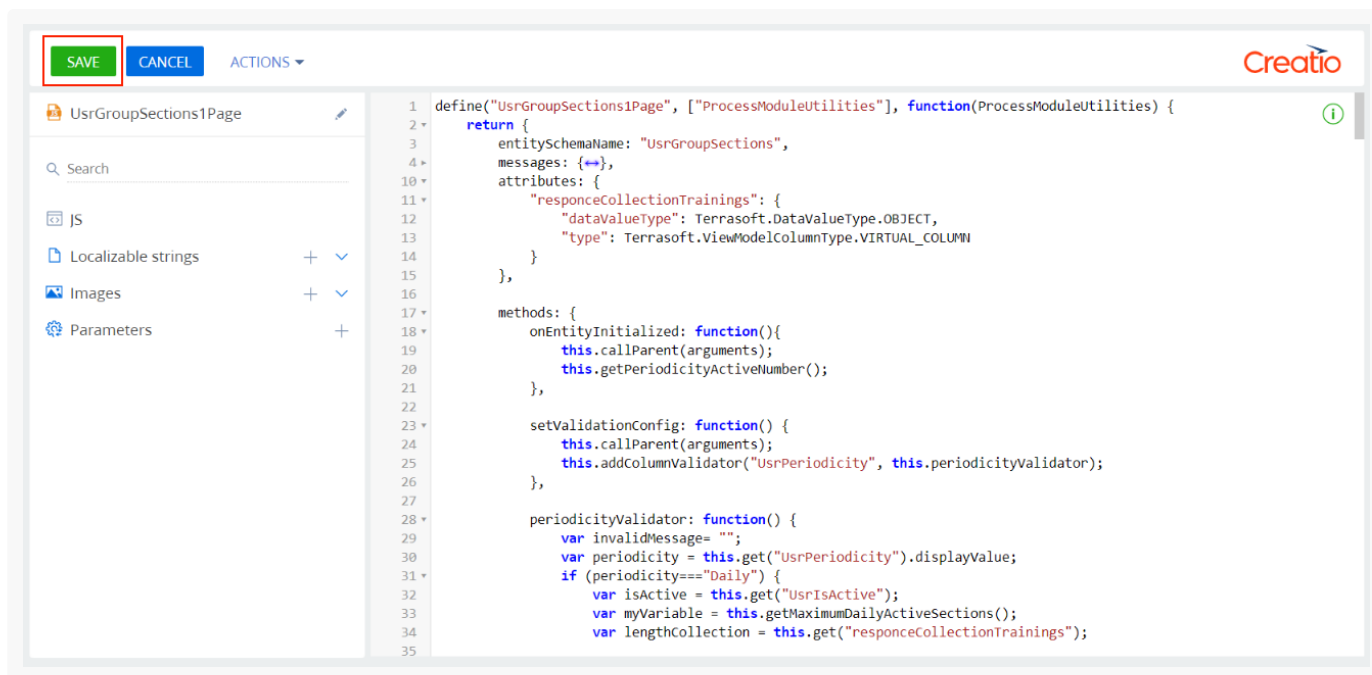
}/**SCHEMA_DETAILS*/,
businessRules: /**SCHEMA_BUSINESS_RULES*/{}/**SCHEMA_BUSINESS_RULES*/,
dataModels: /**SCHEMA_DATA_MODELS*/{}/**SCHEMA_DATA_MODELS*/,
/* Без изменений. */
diff: /**SCHEMA_DIFF*/[

// ...

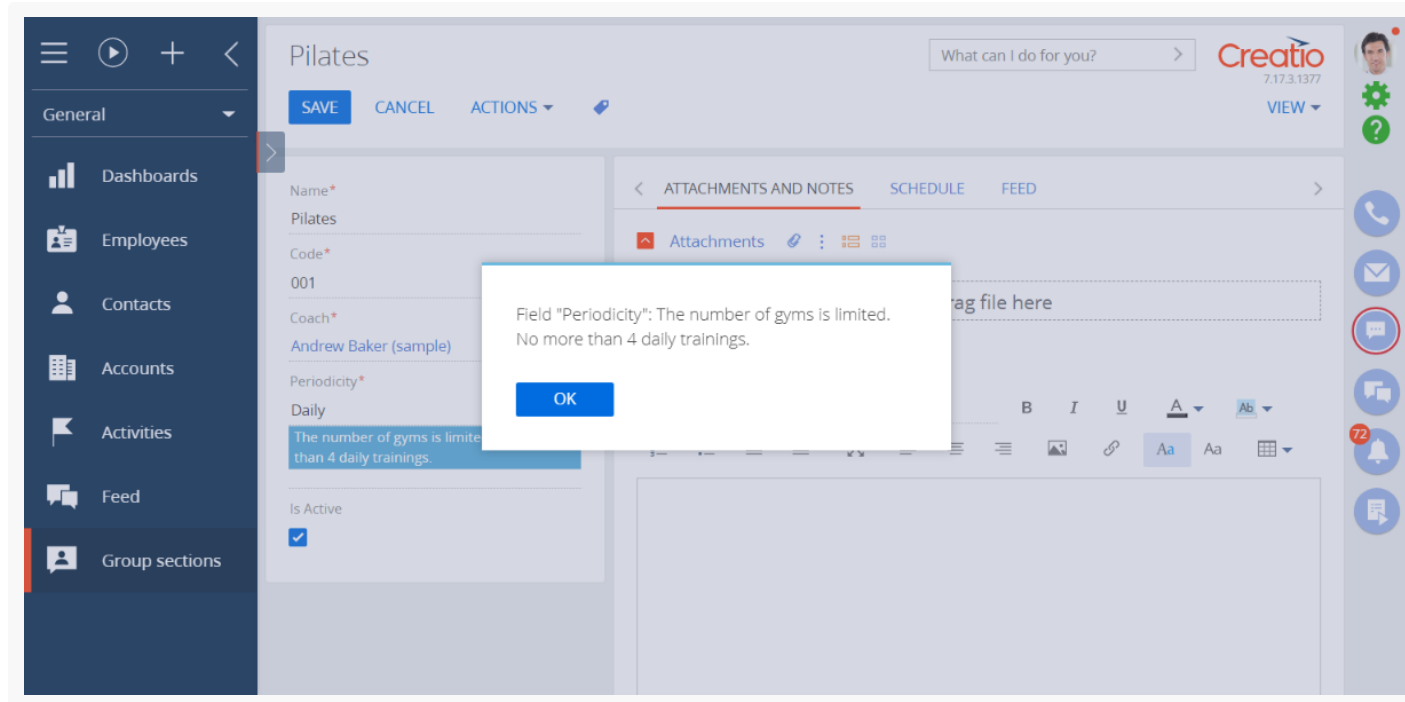
]/**SCHEMA_DIFF*/
};
});

```

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



Результат внесенных изменений:



В результате мы реализовали необходимую бизнес-логику страницы. На [следующем шаге](#) необходимо реализовать автоматическое заполнение расписания групповых занятий.

Шаг 4. Реализуйте автозаполнение расписания

Основы

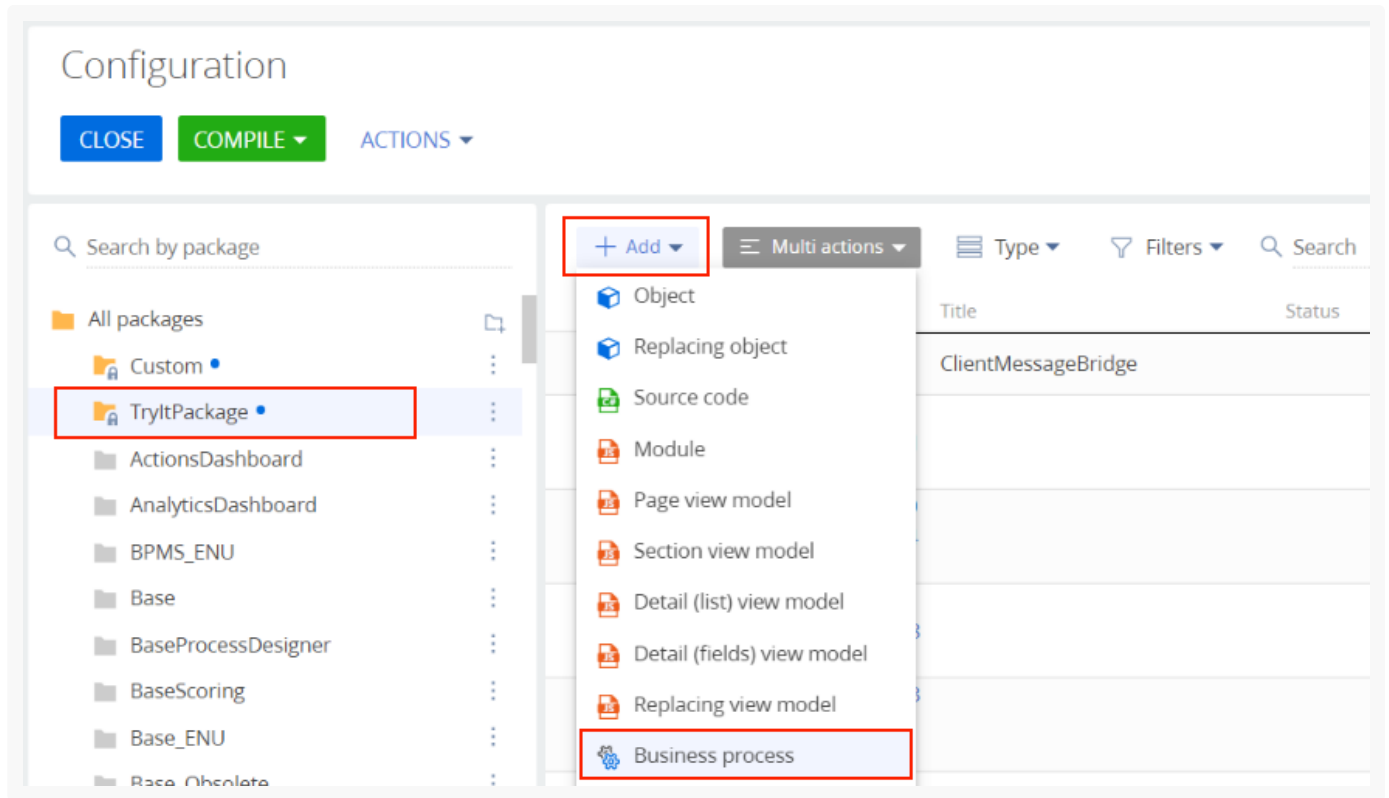
На [предыдущем шаге](#) мы добавили к странице проверку соответствия текущего количества активных ежедневных секций количеству залов для групповых занятий.

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.

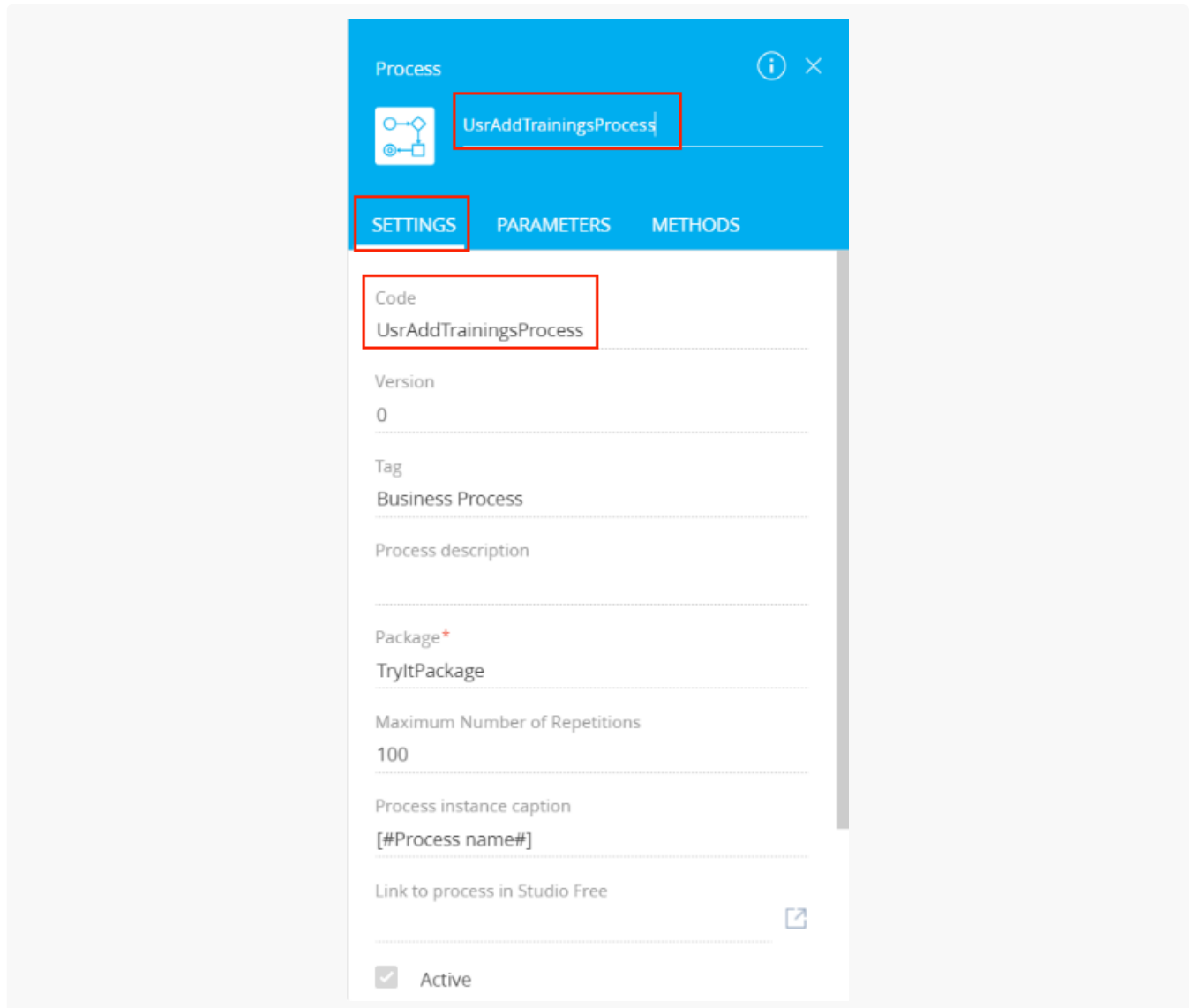
Создать бизнес-процесс

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

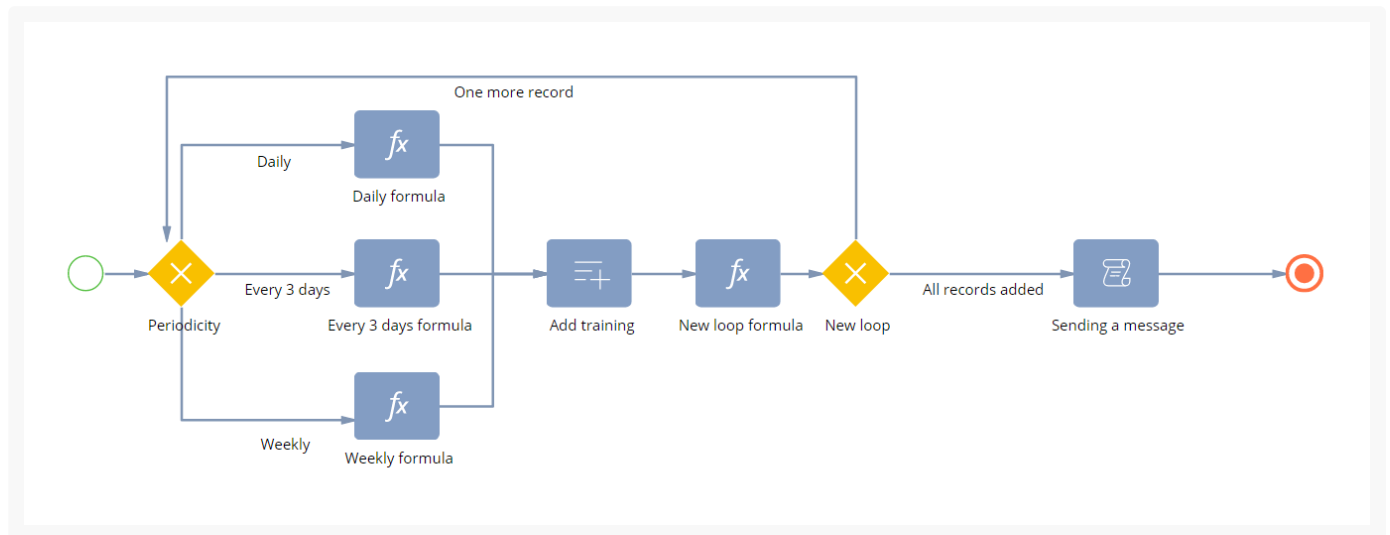
Number

↔ 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:

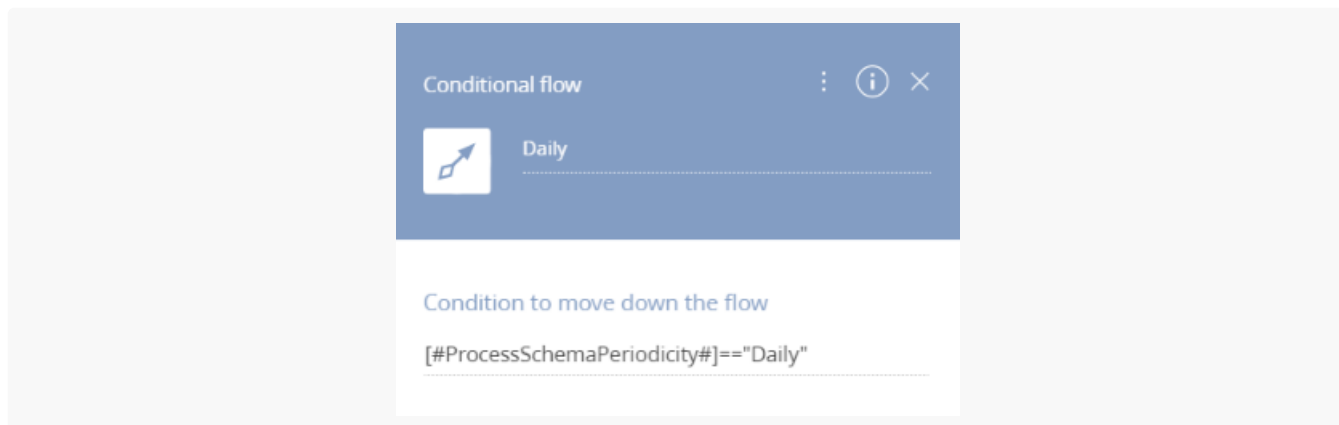
- Логический оператор [*Исключающее ИЛИ*] ([*Exclusive gateway (OR)*] "Periodicity" — проверяет периодичность секции, для которой нужно заполнить расписание, и разветвляет процесс на следующем шаге в зависимости от периодичности.
- Элементы [*Формула*] ([*Formula*]) — процесс выполняет расчет даты занятия, 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".

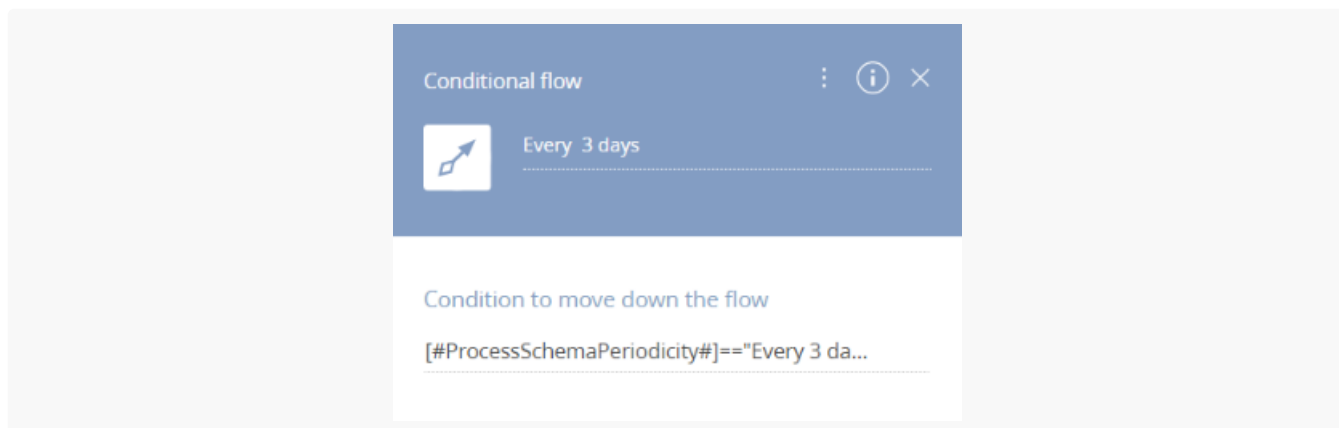
a. [*Условный поток*] ([*Conditional flow*] "Daily":

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



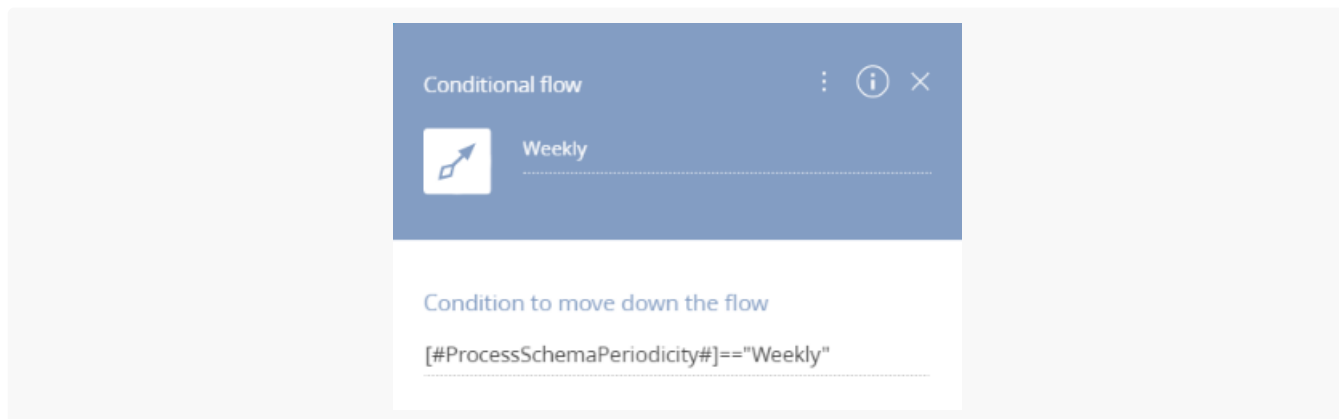
c. [Условный поток]([Conditional flow]) "Every 3 days":

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



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

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

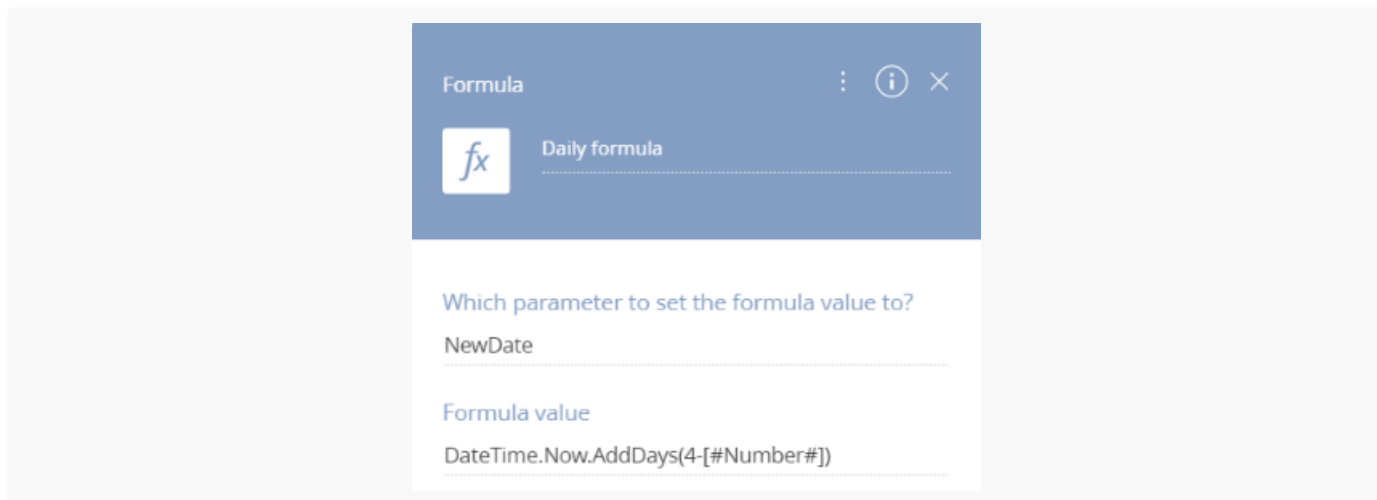


[Формула]([Formula]) "Daily formula":

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

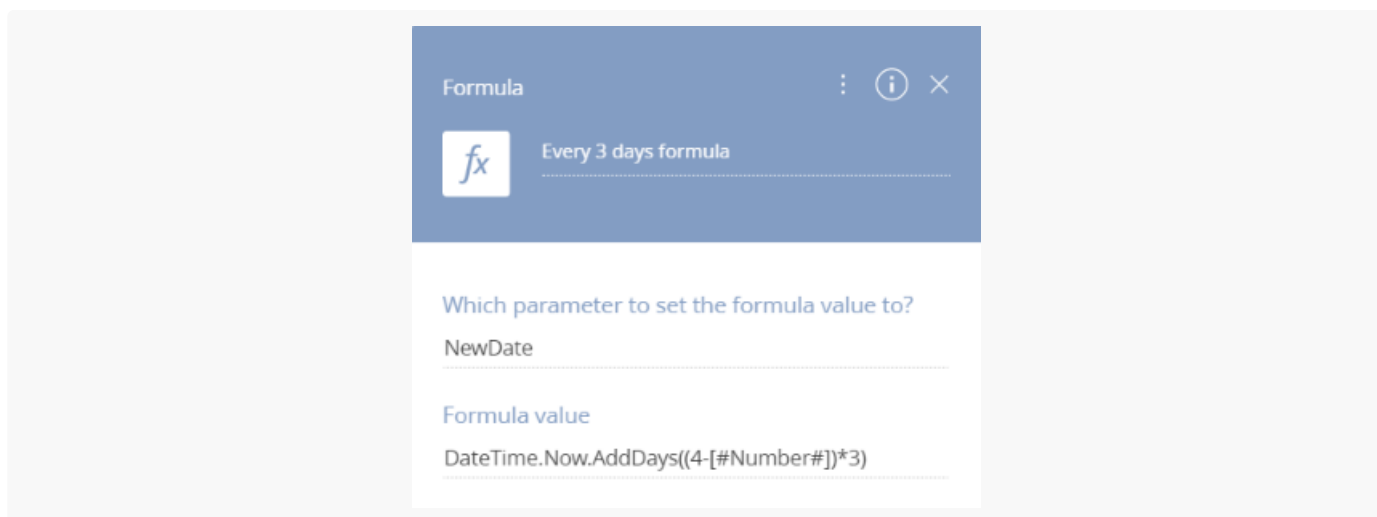
"NewDate".

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



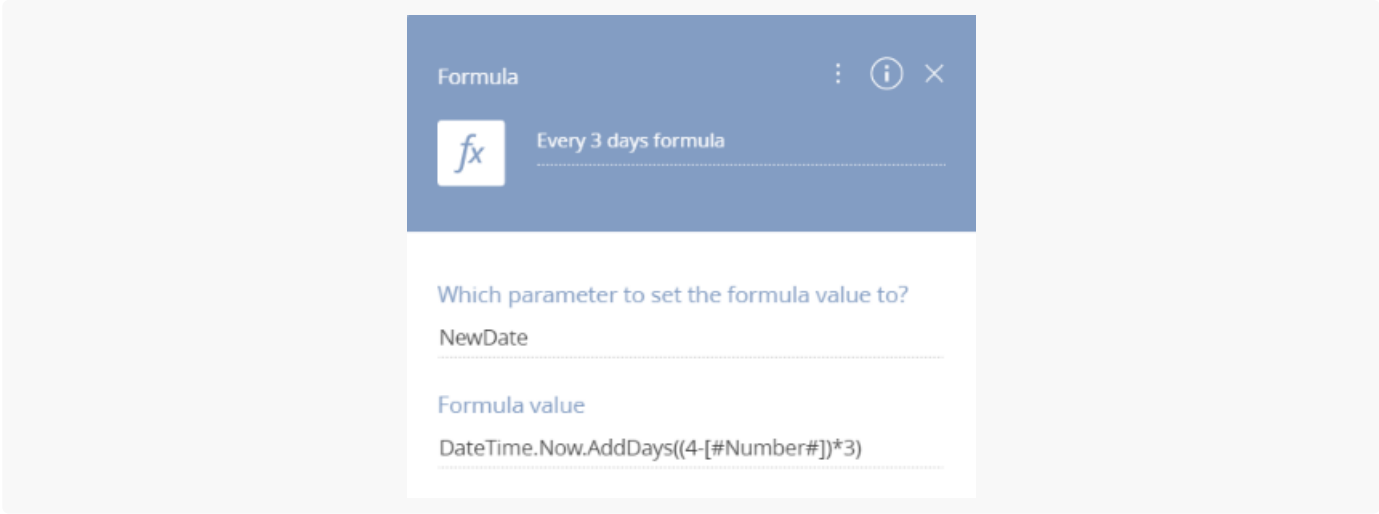
[**Формула**] ([**Formula**]) "**Every 3 days formula**":

- [*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)` .



[**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":

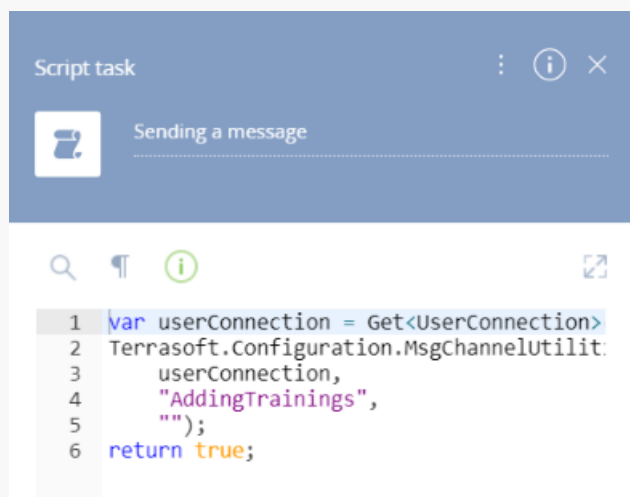
- [Условие перехода]([Condition to move down the flow]) — `[#Number#]>=1`.

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.

Шаг 5. Создайте пользовательский веб-сервис

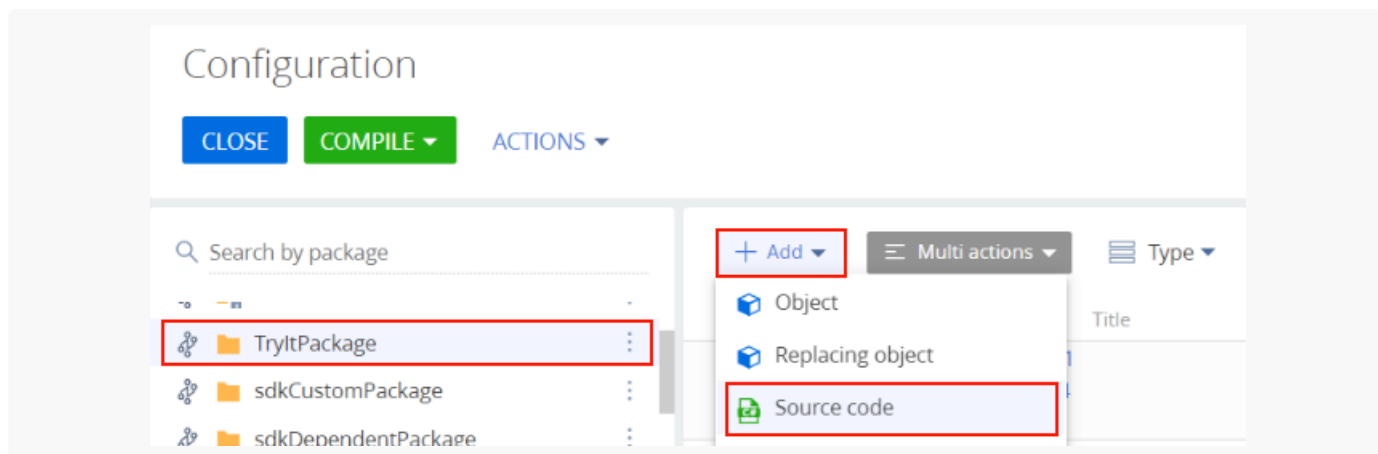
Основы

На [предыдущем шаге](#) мы реализовали автоматическое заполнение расписания групповых занятий.

Теперь реализуем пользовательский веб-сервис, который будет предоставлять информацию о количестве занятий в расписании. Для вызова веб-сервиса добавим кнопку [*Web service*] на страницу записи.

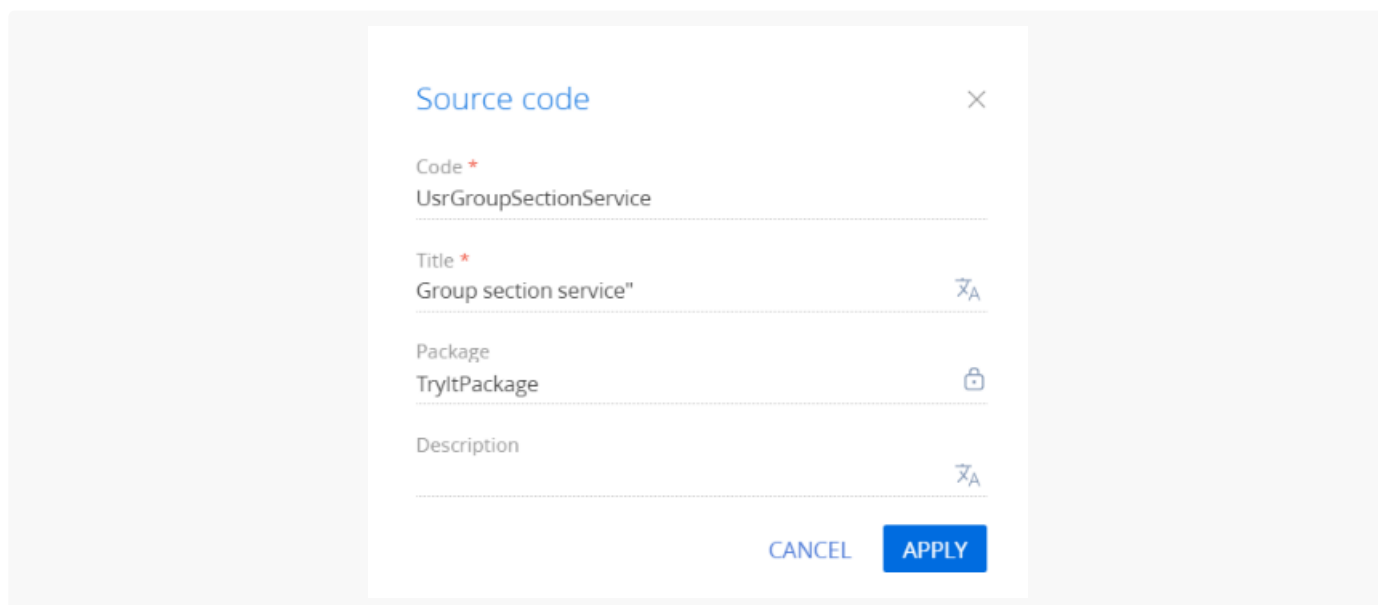
Создать пользовательский веб-сервис

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 [*Исходный код*] ([*Source code*]).



4. Заполните поля свойств схемы значениями:

- [Код] ([Code]) — "UsrGroupSectionService".
- [Заголовок] ([Title]) — "Group section service".



5. В дизайне схем добавьте исходный код.

UsrGroupSectionService.cs

```
namespace Terrasoft.Configuration.UsrGroupSectionService
{
    using System;
    using System.ServiceModel;
    using System.ServiceModel.Web;
    using System.ServiceModel.Activation;
    using Terrasoft.Core;
    using Terrasoft.Core.DB;
    using Terrasoft.Common;
    using Terrasoft.Web.Common;
```

```

using Terrasoft.Core.Entities;

[ServiceContract]
[AspNetCompatibilityRequirements(RequirementsMode = AspNetCompatibilityRequirementsMode.R
public class UsgroupSectionService: BaseService
{

    [OperationContract]
    [WebInvoke(Method = "POST", RequestFormat = WebMessageFormat.Json, BodyStyle = WebMes
    ResponseFormat = WebMessageFormat.Json)]
    public int GetTrainingsQuantity(string code) {
        var groupSectionQuery = new Select(UserConnection)
            .Column("Id")
            .From("UsgroupSections")
            .Where("UsgroupCode")
                .IsEqual(Column.Parameter(code))
            as Select;
        Guid id = groupSectionQuery.ExecuteScalar<Guid>();
        if (id==Guid.Empty) {
            return -1;
        }
        var countQuery = new Select(UserConnection)
            .Column(Func.Count("Id")).As("Count")
            .From("UsgroupTrainings")
            .Where("UsgroupSectionsId")
                .IsEqual(Column.Parameter(id))
            as Select;
        int result = countQuery.ExecuteScalar<int>();

        return result;
    }
}

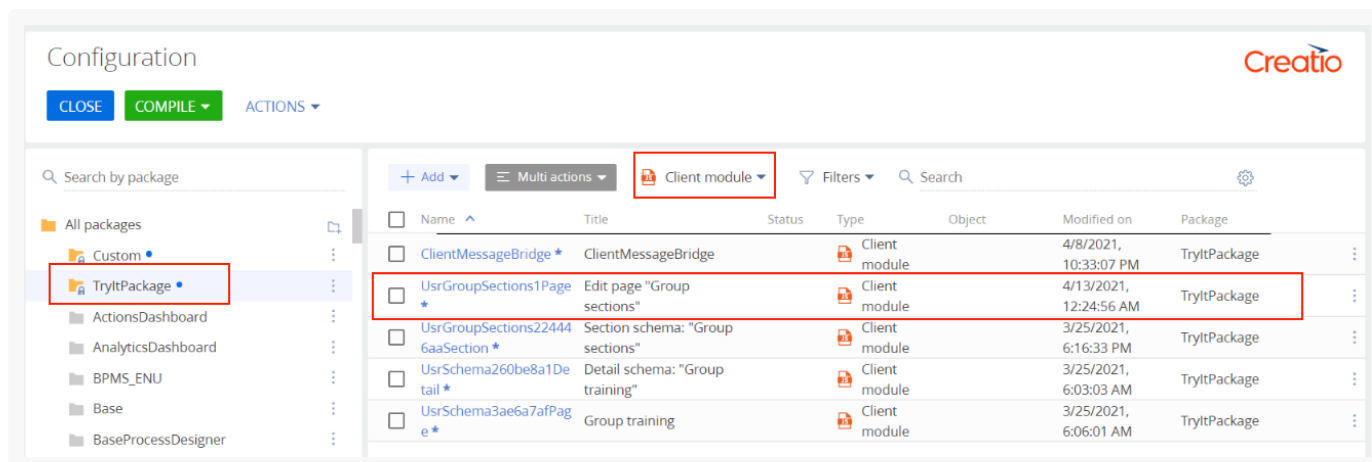
```


6. Сохраните схему, нажав на кнопку [*Опубликовать*] ([*Publish*]).

Отредактировать исходный код страницы

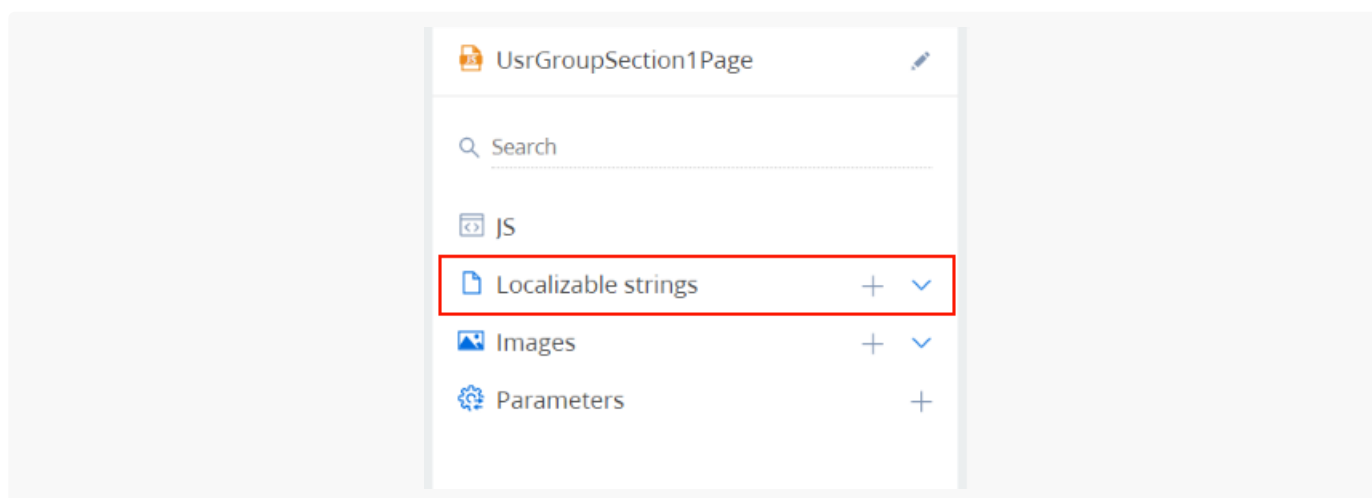
На страницу групповой секции добавьте кнопку, которая будет вызывать веб-сервис и выводить количество занятий в расписании во всплывающем окне.

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 только схемы с типом [*Клиентский модуль*] ([*Client module*]).



4. Двойным нажатием мыши откройте схему `UsrGroupSections1Page`.
5. Добавьте в схему новую локализуемую строку, которая будет использоваться для названия кнопки. В блоке [*Локализуемые строки*] ([*Localizable strings*]) панели свойств нажмите кнопку  и заполните **свойства локализуемой строки**:

- [*Код*] ([*Code*]) — "GetServiceInfoButtonCaption".
- [*Значение*] ([*Value*]) — "WebService".



6. Отредактируйте исходный код:

`UsrGroupSections1Page.js`

```
define("UsrGroupSections1Page", ["ServiceHelper", "ProcessModuleUtilities"], function(Service
return {
  entitySchemaName: "UsrGroupSections",
  /* Без изменений. */
  messages: {
    // ...
  },

  /* Без изменений. */
```

```

attributes: {
    // ...
},

modules: /**SCHEMA_MODULES*/{}/**SCHEMA_MODULES*/,

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

businessRules: /**SCHEMA_BUSINESS_RULES*/{}/**SCHEMA_BUSINESS_RULES*/,

/* Добавляем новый метод к существующим. */
methods: {

    // ...

    /* Метод-обработчик нажатия кнопки. */
    onGetServiceInfoClick: function() {
        /* Получаем код секции, для передачи в качестве входящего параметра метода сервиса. */
        var code = this.get("UserCode");
        var serviceData = {
            code: code
        };
        /* Вызов метода сервиса. */
        ServiceHelper.callService("UsrGroupSectionService", "GetTrainingsQuantity",
            function(response) {
                var result = response.GetTrainingsQuantityResult;
                /* Отображение результата выполнения метода сервиса во всплывающем окне. */
                this.showInformationDialog(result);
            }, serviceData, this);
    }
},

dataModels: /**SCHEMA_DATA_MODELS*/{}/**SCHEMA_DATA_MODELS*/,

/* Добавляем отображение кнопки на странице записи. */
diff: /**SCHEMA_DIFF*/[

    // ...
    /* Добавляем элемент кнопка на страницу записи. */
    {
        "operation": "insert",
        /* Имя родительского элемента. */
        "parentName": "ActionButtonsContainer",
        "propertyName": "items",
        /* Имя элемента. */
        "name": "GetServiceInfoButton",
        "values": {

```

```

/* Тип элемента - кнопка. */
itemType: Terrasoft.ViewItemType.BUTTON,
/* Заголовок элемента получаем из локализованной строки. */
caption: {bindTo: "Resources.Strings.GetServiceInfoButtonCaption"},
/* Метод-обработчик нажатия кнопки. */
click: {bindTo: "onGetServiceInfoClick"},
enabled: true,
/* Позиция кнопки на странице. */
"layout": {"column": 1, "row": 6, "colSpan": 2, "rowSpan": 1}
}
},

]**SCHEMA_DIFF*]
};
});

```

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

В результате мы реализовали веб-сервис, который возвращает количество занятий групповой секции.

