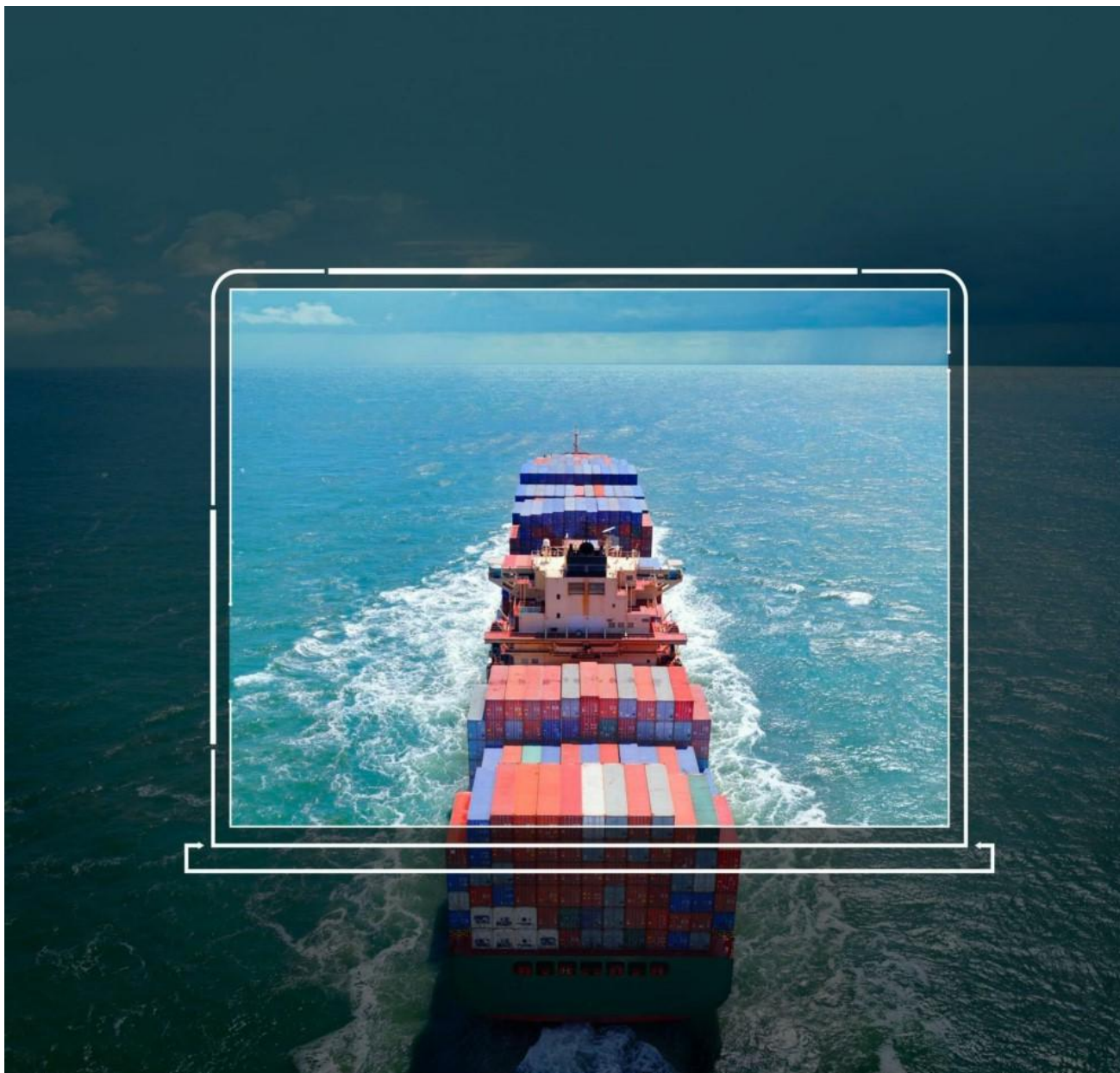




Mendix 敏捷开发零基础学习

Mendix 敏捷开发零基础学习《一》- 基础 1

Mendix 敏捷开发零基础学习《二》-进阶 35

Mendix 敏捷开发零基础学习《三》 143

Mendix 敏捷开发零基础学习《一》- 基础

数据模型、页面创建、数据源绑定、属性绑定、关系型数据模型

- 一、 环境依赖
- 二、 业务场景
- 三、 需求分析
- 四、 项目实施
 1. 创建项目
 2. 创建数据模型
 3. 创建页面
 4. 数据源绑定
 5. 属性绑定
 6. 数据新增
 7. 数据删除
 8. 关系型数据模型

一、环境依赖

1. **Mendix 账号：** 需要用公司邮箱注册
注册地址： <https://signup.mendix.com/link/signup/?ref=158899>
2. **开发工具：** MendixStudioPro 9.0.5
下载链接： <https://pan.baidu.com/s/1AWDXVT0jRwowZKWBZzLD0g> 提取码：SIMX
3. **系统要求：** win10
4. **安装 JDK：** OpenJDK11U-jdk_x64_windows_hotspot_11.0.3_7
下载连接： <https://pan.baidu.com/s/1PE4GBkBoL-8Wjx-mvelzVA> 提取码：4cqt
5. **安装.NET 环境：** Microsoft .NET Framework 4.7.2
下载连接： <https://pan.baidu.com/s/12LQqGpMFVKzyNcrUZGkinw> 提取码：hosx
6. **安装 C++环境：** Microsoft Visual C++ 2010 SP1 Redistribution Package
下载连接： <https://pan.baidu.com/s/11gVjinWdhY3J1XfX9VrPvg> 提取码：sygg

二、业务场景

现有一学校，需要记录课程信息（课程信息里边包含该课程需要几天、价钱、说明等）、教师信息(姓名、邮箱等)、课程位置（标题、地址等）、报名人员（名称、地址、邮箱等）、课程培训活动等，方便维护系统。

三、需求分析

根据业务场景分析, 我们需要创建五个数据模型 (课程 Courses、教师 Teacher、地址 Location、报名人员 Trainee、课程培训活动 TrainingEvent), 分别来存储这些信息。下边列举一下具体的属性有哪些:

课程 Courses: [【返回 Entity 数据表配置】](#)

属性	备注	类型
Title	标题	String
Description	说明	String
Duration	需要多久/天	Integer
Price	价钱	Decimal

教师 Teacher:

属性	备注	类型
Name	名称	String
EmailAddress	邮箱地址	String

地址 Location:

属性	备注	类型
Name	名称	String
Address	地址	String

报名人员 Trainee:

属性	备注	类型
Name	名字	String
Address	地址	String
EmailAddress	邮箱地址	String

课程培训活动 TrainingEvent: [【完成 Entity 配置, 跳转到下一步: 创建页面】](#)

(一对多的数据结构, 需要依附于其他的数据表, 会在 [8.关系型数据模型](#) 重点解说如何配置)

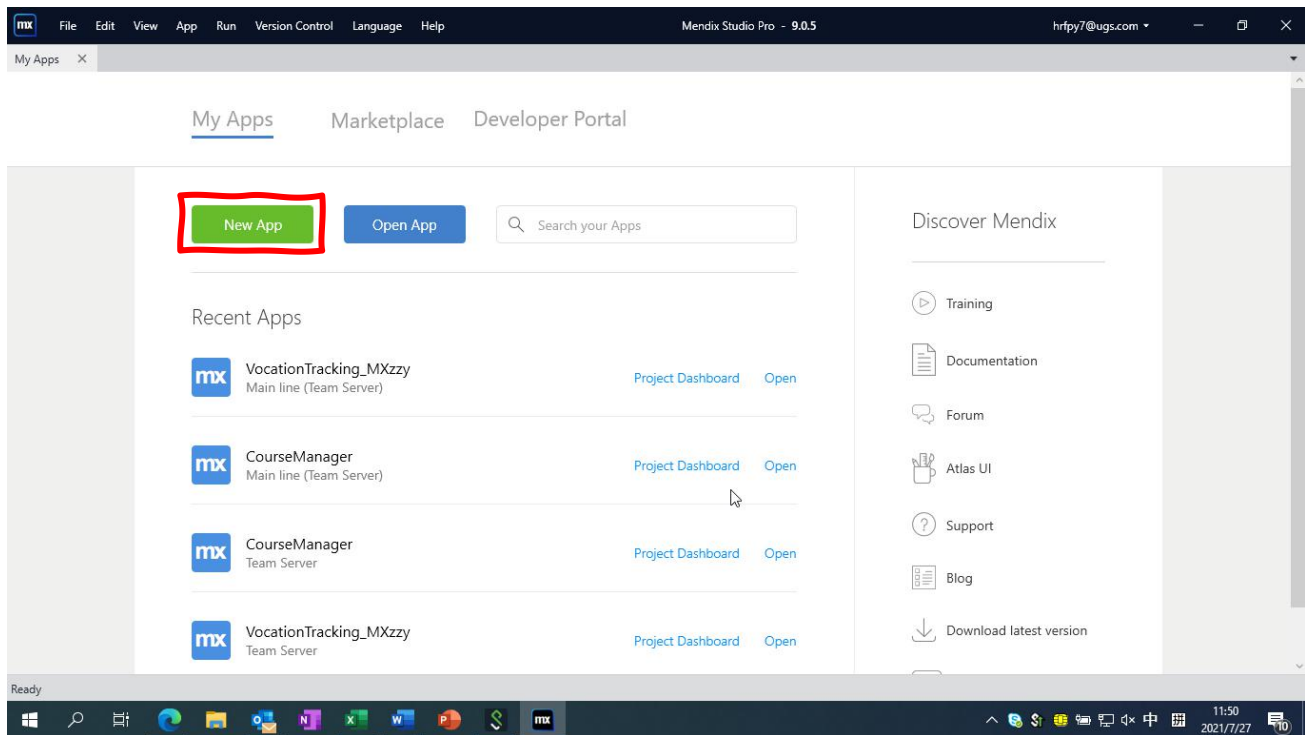
属性	备注	类型
startDate	开始时间	Date and time (这是 Mendix 提供的时间类型)
endDate	结束时间	Date and time (这是 Mendix 提供的时间类型)

四、项目实施

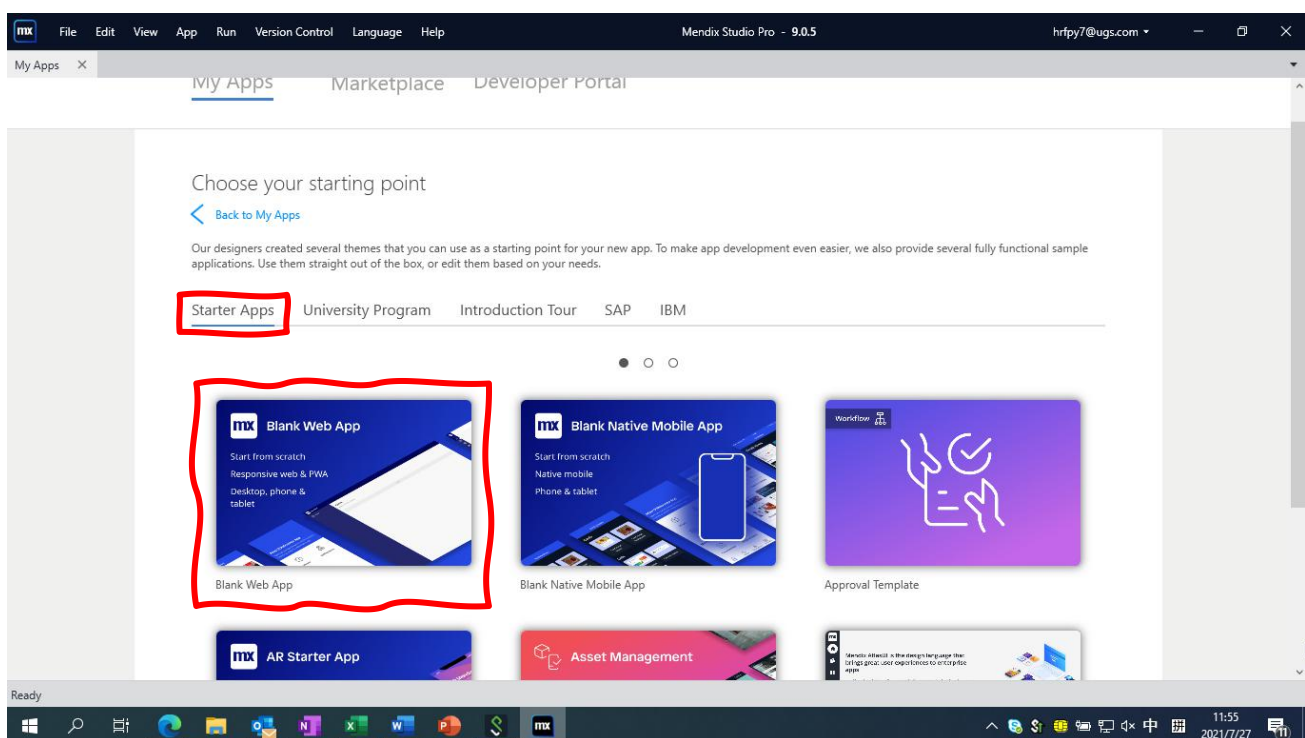
根据以上分析结果，我们可以根据 Mendix Studio Pro 开发工具实现基础的业务场景，下面开始：

1. 创建项目

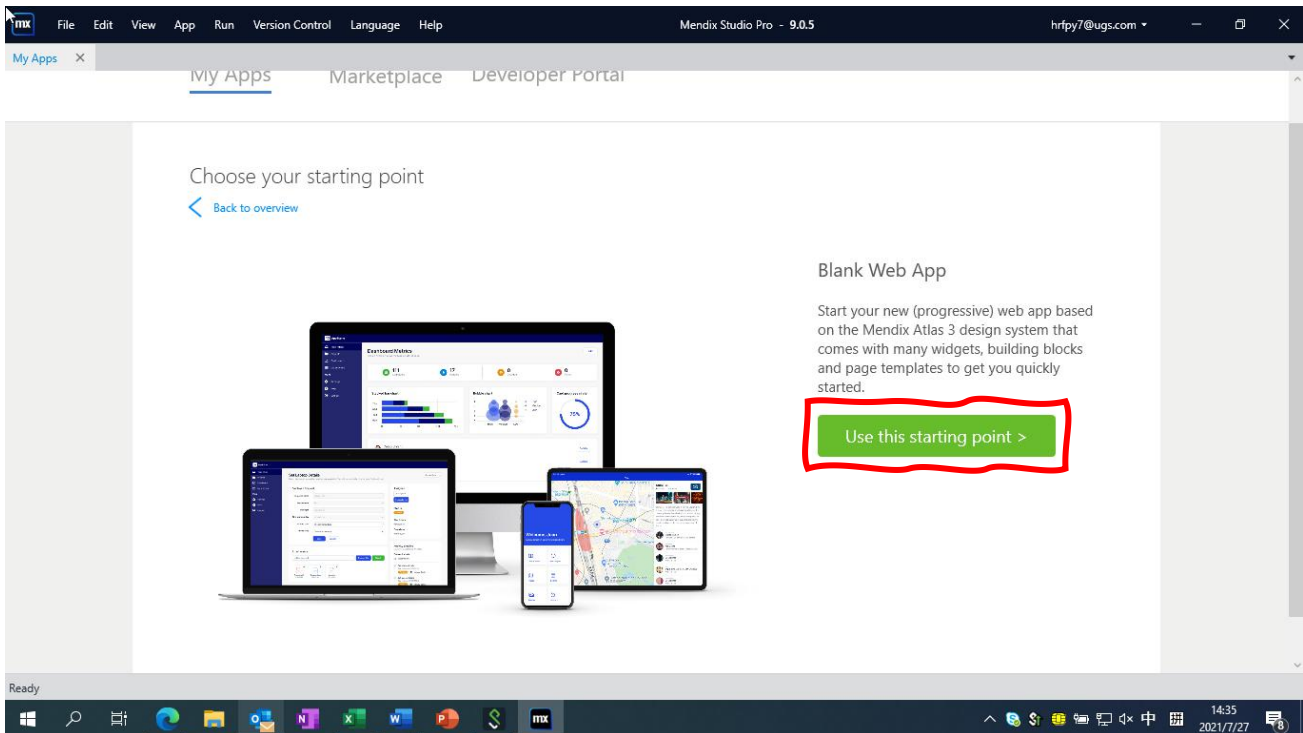
1. 打开 Mendix Studio Pro 新创建一个空白项目，选择 **New App** 按钮。



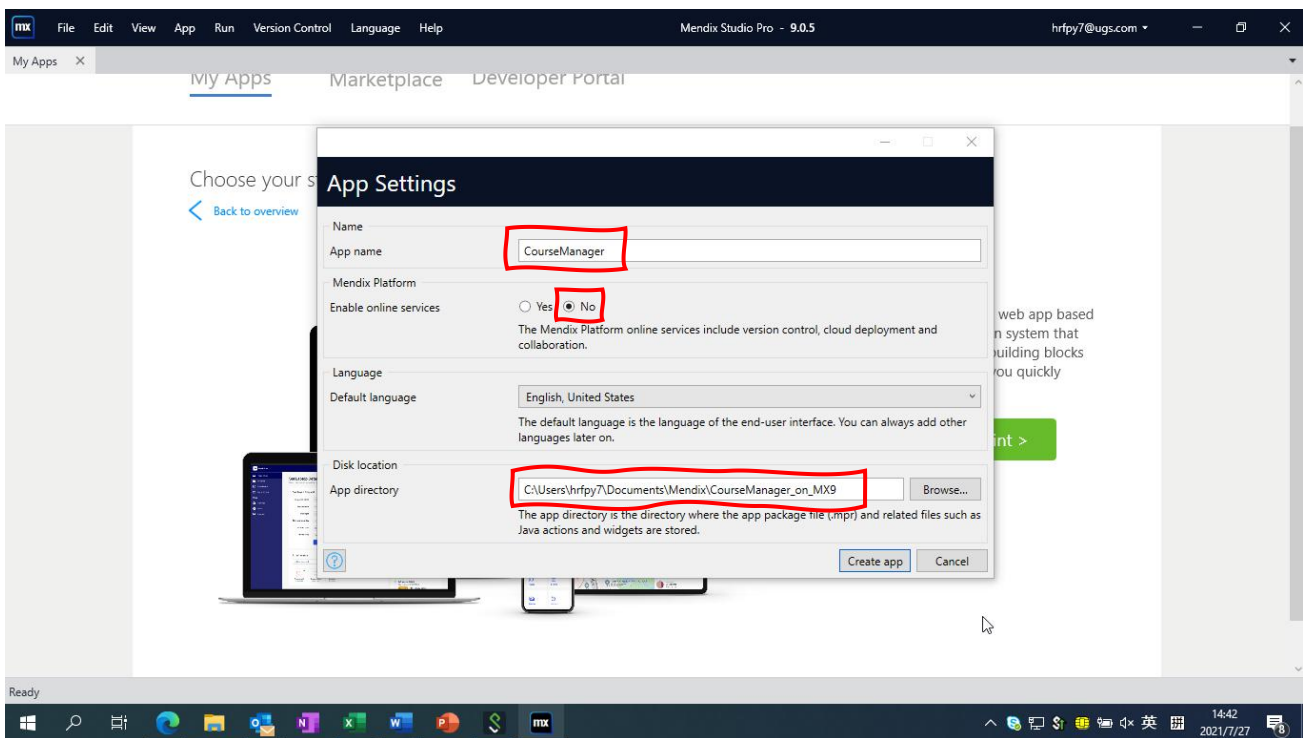
2. 在 **Starter Apps** 中，选择 **Blank Web App** 模版。（这里有很多模版，有机会可以试试。）



3. 点击 **Use this starting point**, 使用当前模版

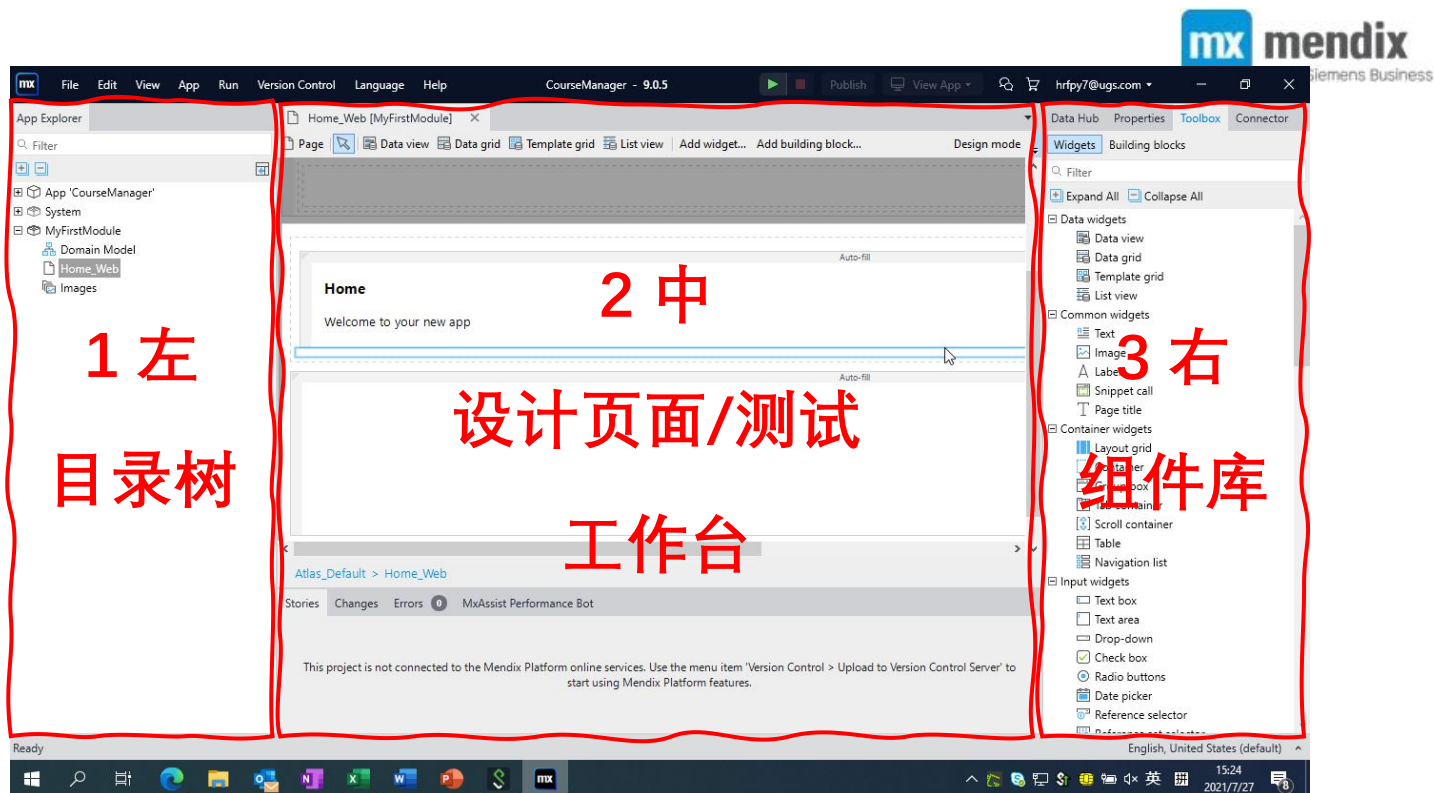


4. 在 **App Name** 中填写项目名称 **CourseManager**, **Enable Online Services** 选择 **No** (该选项是否需要联机服务, 联机服务会很慢, 所以选择 NO。我们在本地也不需要联机服务!), 在 **Project Directory** 选择保存 app 项目文件的路径 (可自定义)。一般情况下只需要填写这三项内容即可。最后点击 **Create app** 完成创建项目工程。



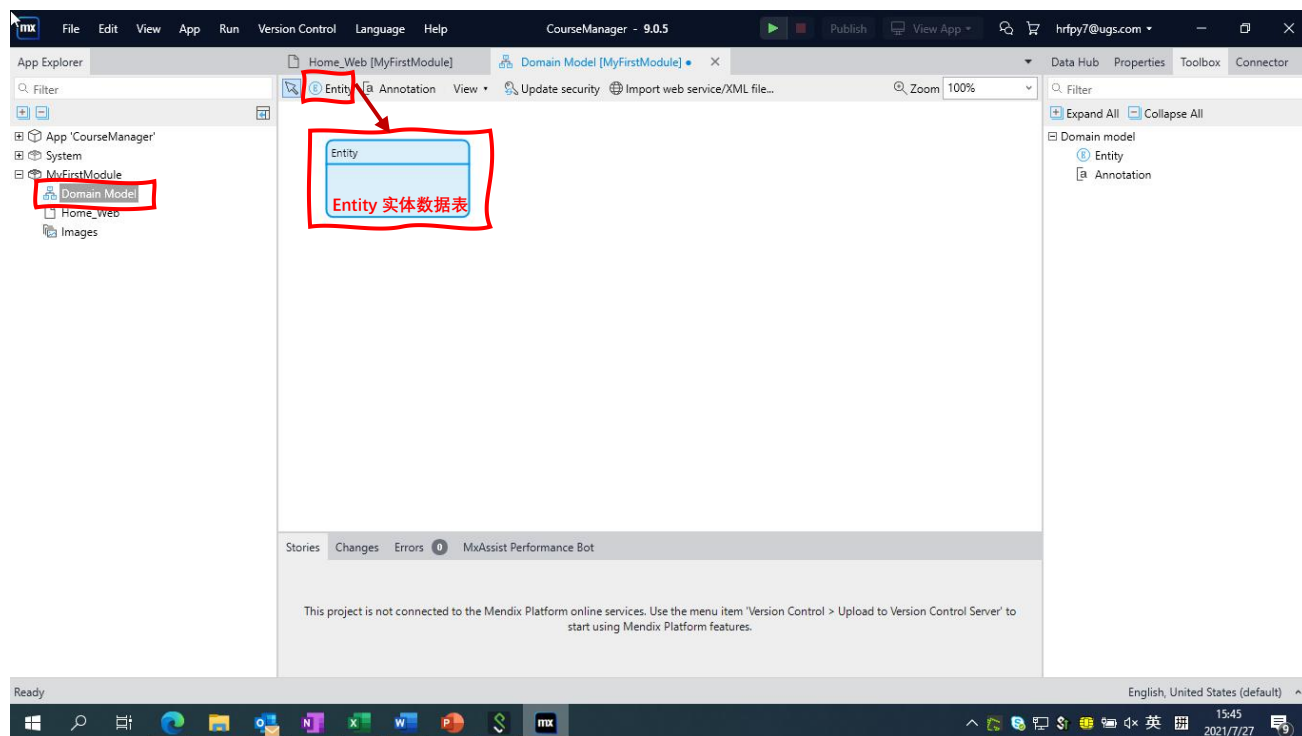
5. 我们新创建的工程, 进来以后有我们常用的三大块:

1 左是页面的目录树。2 中是设计页面的工作台。3 右是常见的一些组件: 如按钮、表格等。

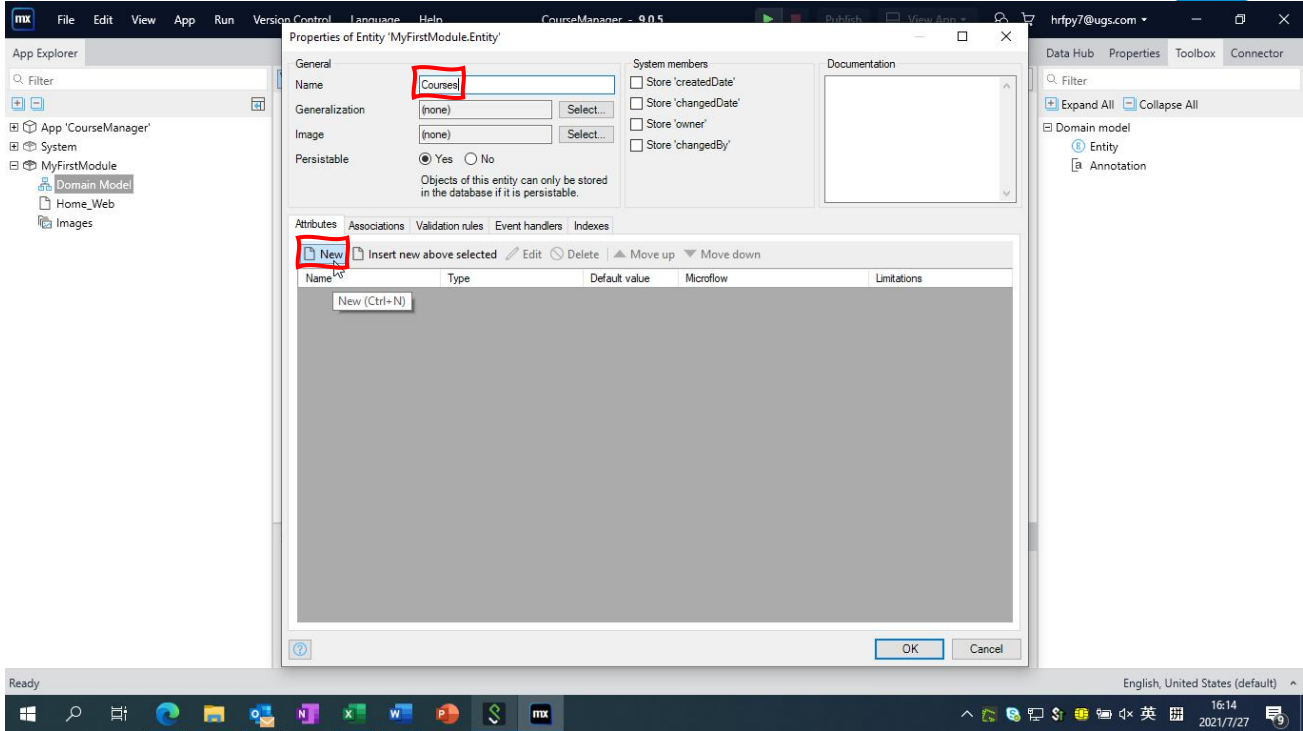


2. 创建数据模型

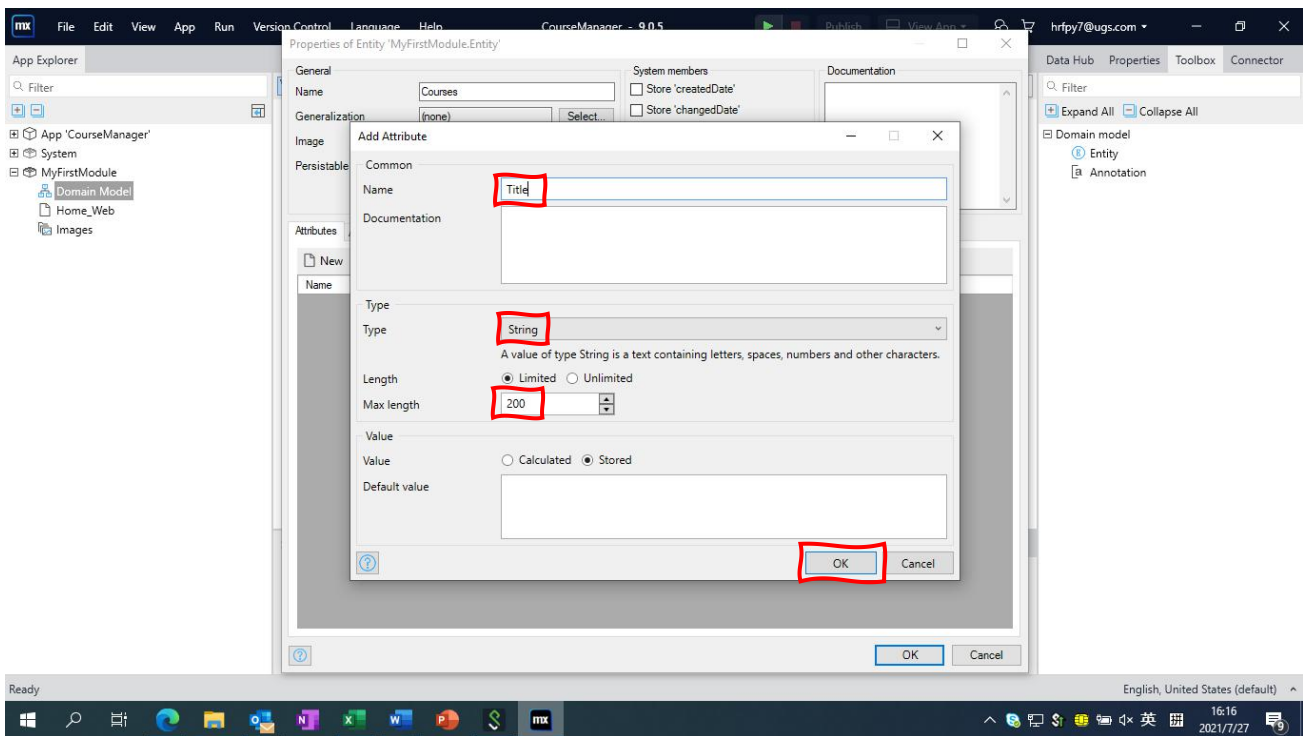
1. 创建数据模型：首先，双击目录树(1 左)中的 **DomainModel**，在页面(2 中)点击 **Entity** 按钮，可将 Entity（实体）的数据表拖动至中间白布中任意位置单击，一个默认的实体数据表就生成好了。



2. 双击刚生成的 **Entity** 数据表，在弹出的页面中配置课程 **Courses** 的数据结构：在 **Name** 栏填写实体名称 **Courses**，并在 Attribute（属性）中点击 New 新建实体属性。



3. 在弹出的页面上，**Name** 栏输入属性的名称 **Title**，选择 **Type**（数据类型）为 **String**（字符类型），输入 **MAX length**（最大长度）为 **200**，最后点击 OK。



4. 创建其他的属性，参考数据模型：[课程 Courses](#)（列举 Duration，循环 2-3 步骤，创建方式一样）【在 Attributes 中点击 New->输入 Attribute Name，选择数据类型 Type】

Add Attribute

Common

Name: Duration

Documentation: 需要多久/天

Type

Type: Integer

A value of type Integer is a positive or negative number without decimals. The maximum value that can be represented is +/-2147483647. Use Long if you need to represent larger numbers.

Value

Value: ☐ Calculated ☒ Stored

Default value: 0

OK Cancel

Properties of Entity 'MyFirstModule.Entity'

General

Name: Courses

Generalization: (none) Select...

Image: (none) Select...

Persistable: ☒ Yes ☐ No

Objects of this entity can only be stored in the database if it is persistable.

System members

☐ Store 'createdDate'

☐ Store 'changedDate'

☐ Store 'owner'

☐ Store 'changedBy'

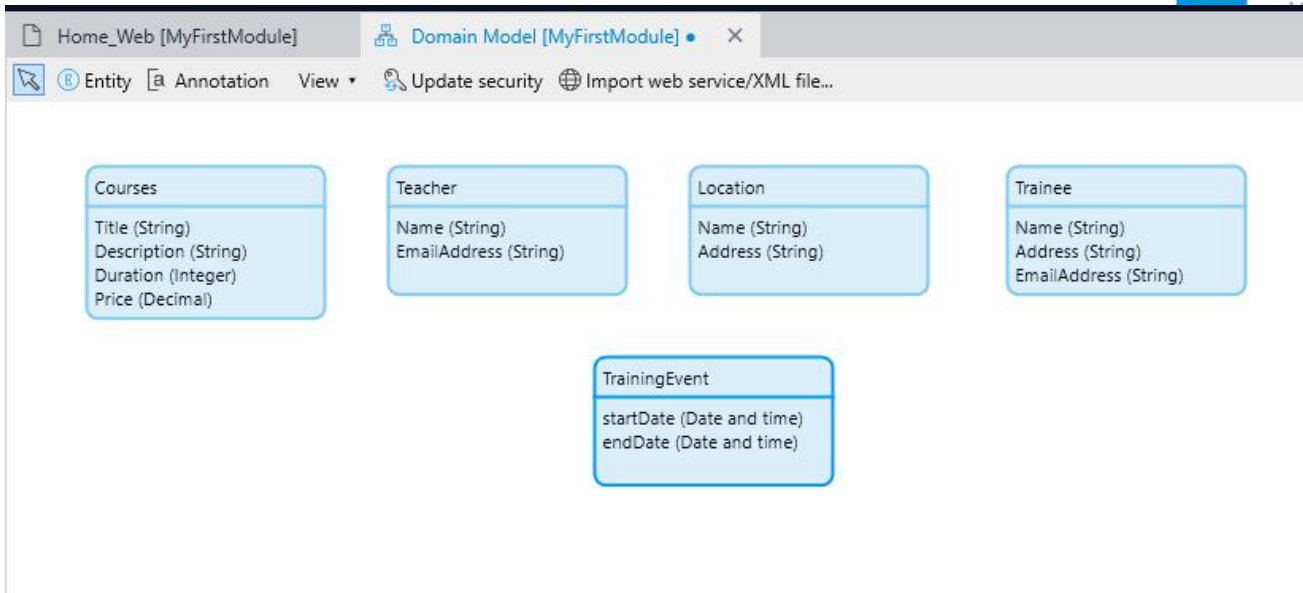
Documentation

Attributes

Name	Type	Default value	Microflow	Limitations
Title	String (200)			
Description	String (200)			
Duration	Integer	0		
Price	Decimal	0		

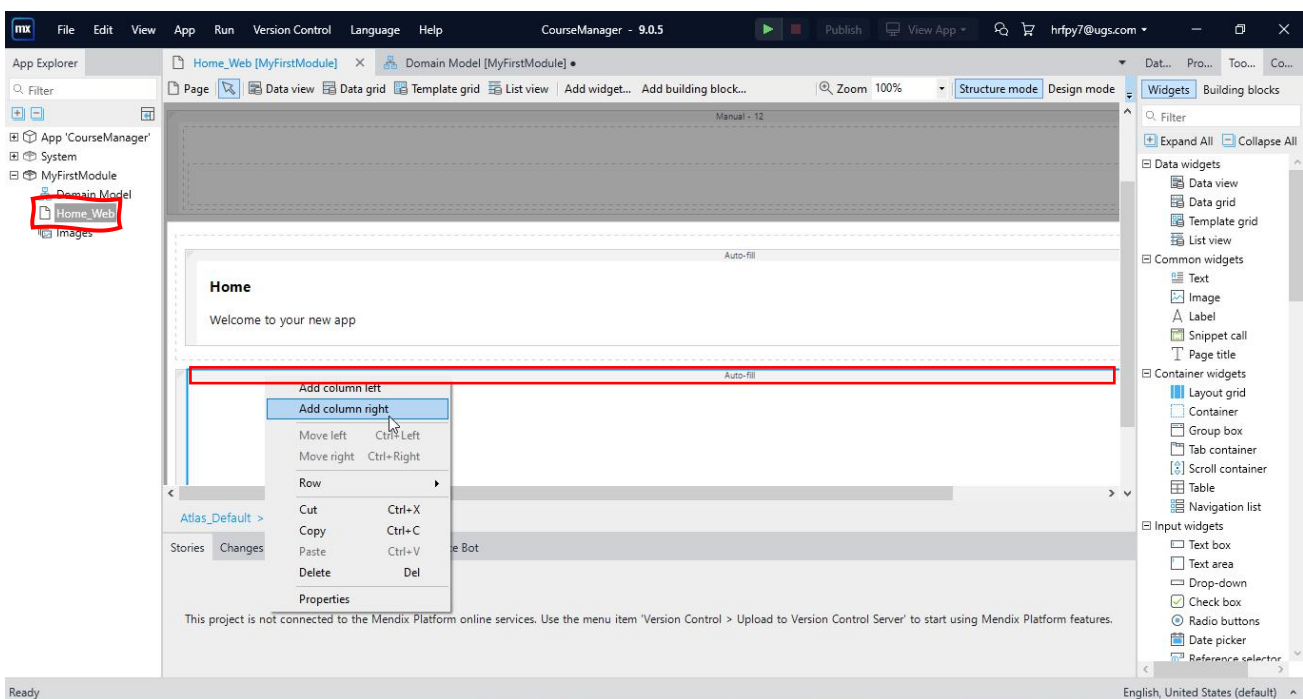
OK Cancel

5. 创建其余数据结构[教师 Teacher](#)、[地址 Location](#)、[报名人员 Trainee](#) 和 [课程培训活动 TrainingEvent](#): 循环 1-3 步骤【New Entity -> 双击 Entity->填写 Entity Name->New Attributes 属性】

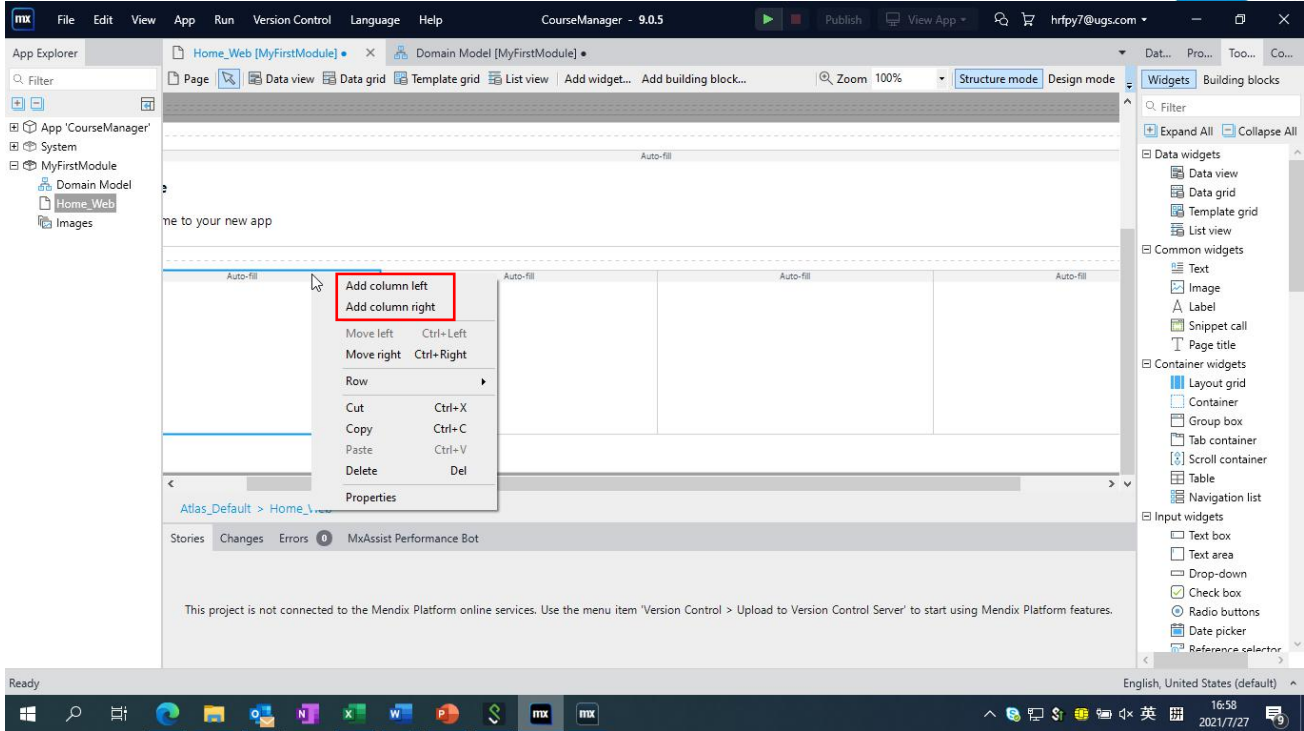


3. 创建页面

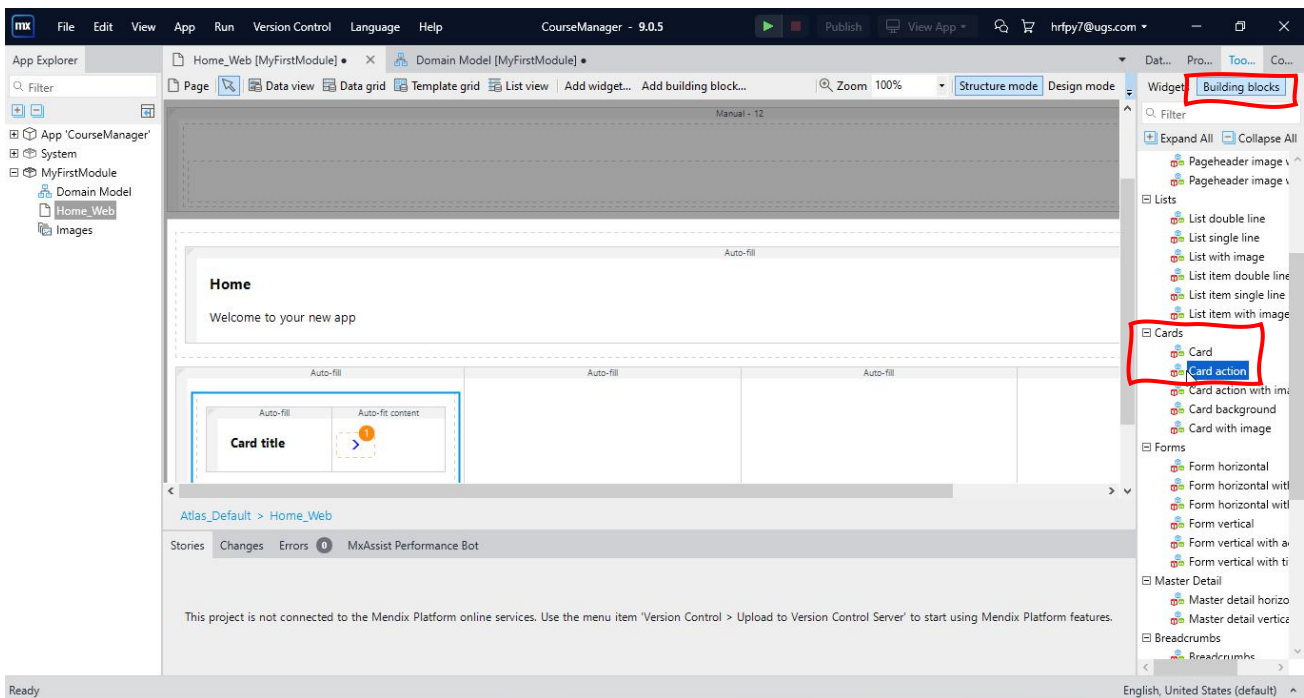
1. 数据结构都定义好了, 现在需要搭建页面程序, 双击左 1 目录树的 **Home_Web** (该页面是我们程序的主入口)。



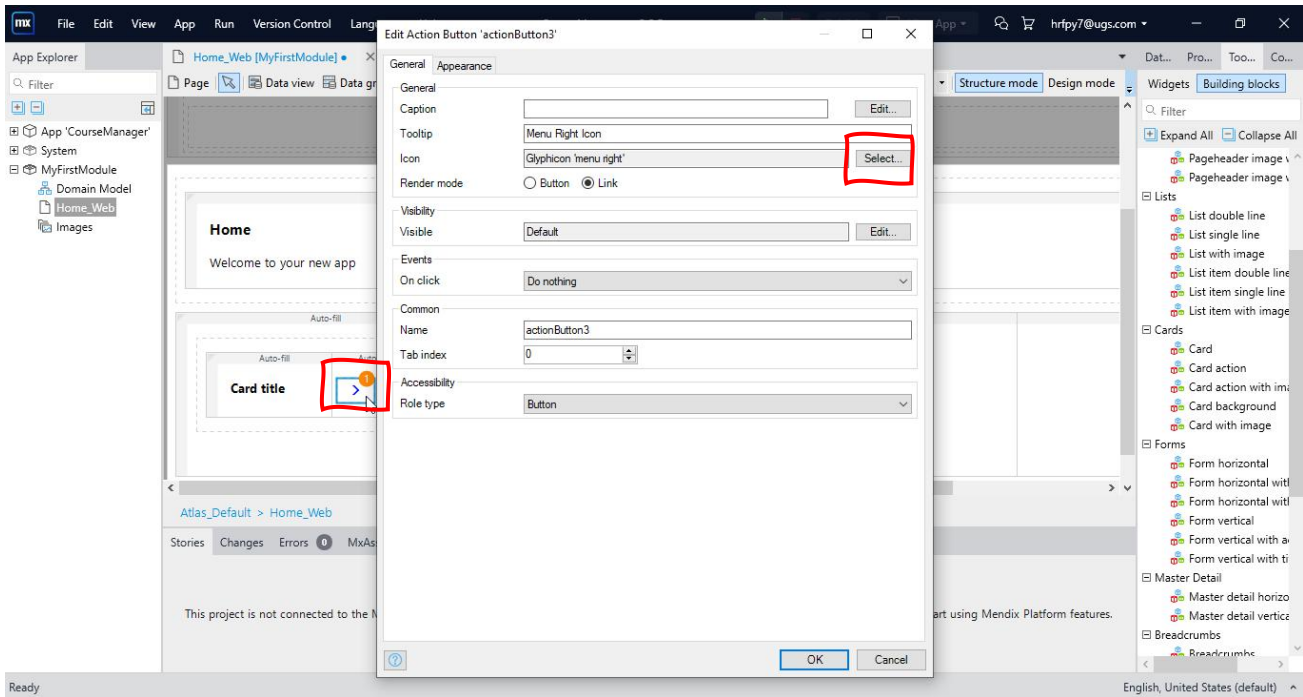
2. 根据业务场景, 首先创建四个部分, 分别是课程 **Courses**、地址 **Location**、教师 **Teacher** 和报名人员 **Trainee**, 分为四列。右击该区域上灰色 Auto-fill 区域, 选择 Add column left/right (都可), 重复 3 次。Mendix 会自动将内容分为四等分。



3. 添加好四列以后，我们在 **3 右** 区域的 **Building blocks** 搜索组件 card，选择 **Card Action** 组件，然后拖入到每一列当中。

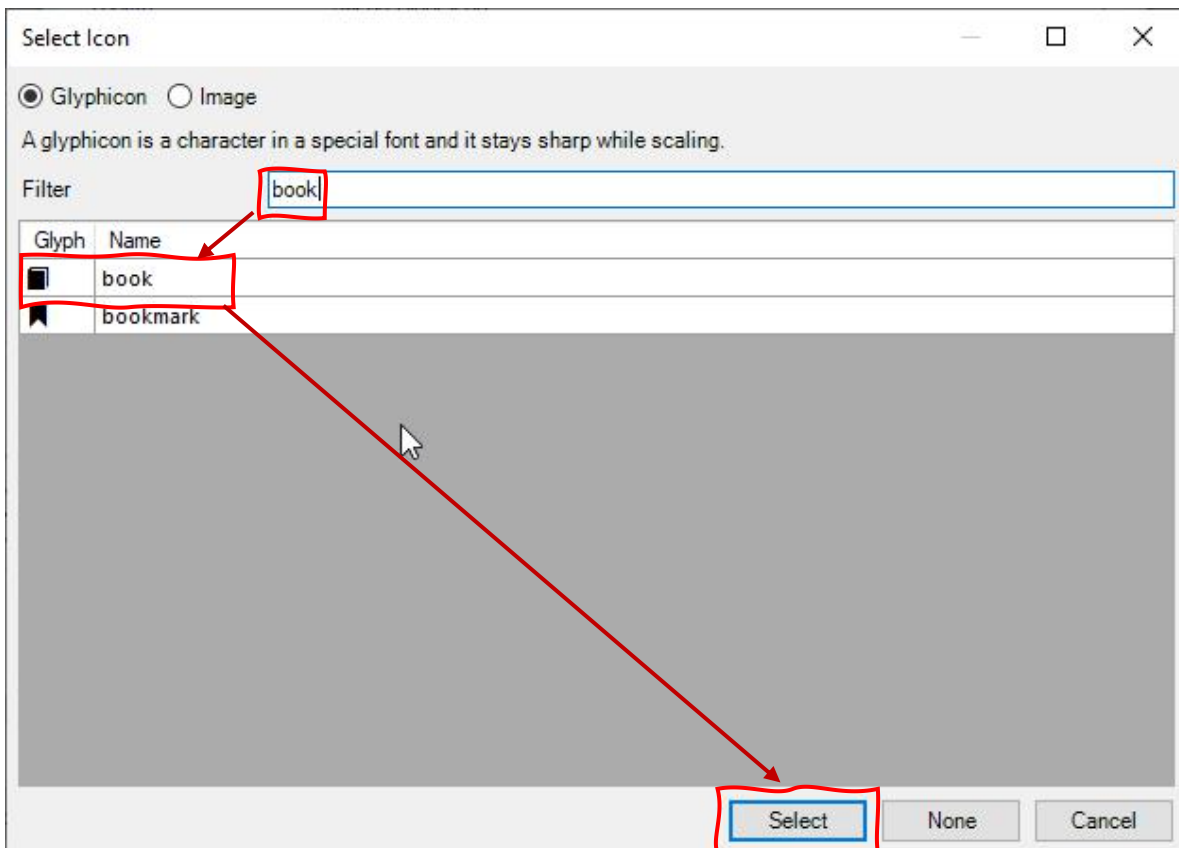


4. 接着更换 card 图标：双击虚线红框，在弹出的页面中点击 Select 按钮。



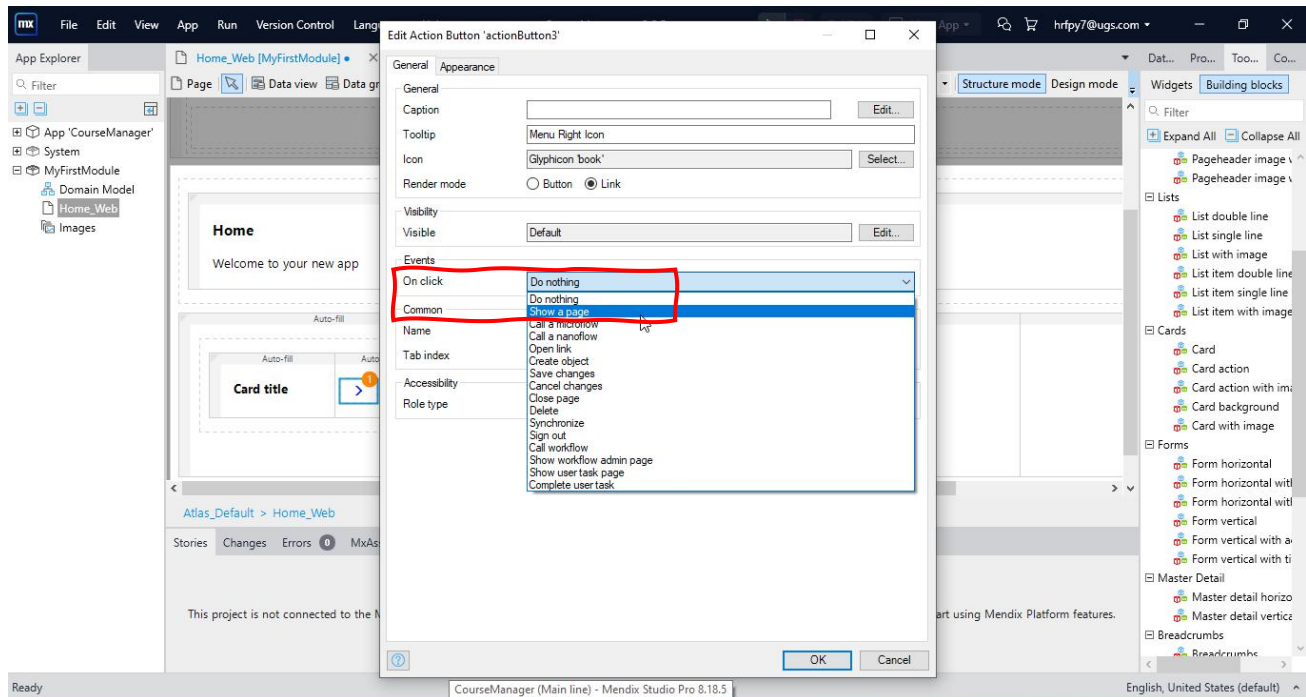
5. 在弹出的页面中搜索 **book**，选择下列图标，然后点击 select 按钮，就可以把当前图标更换为符合我们需求的图标。

【Entity—Icon: Location--map; Teacher--user; Trainee—education; TrainingEvent—Calendar】

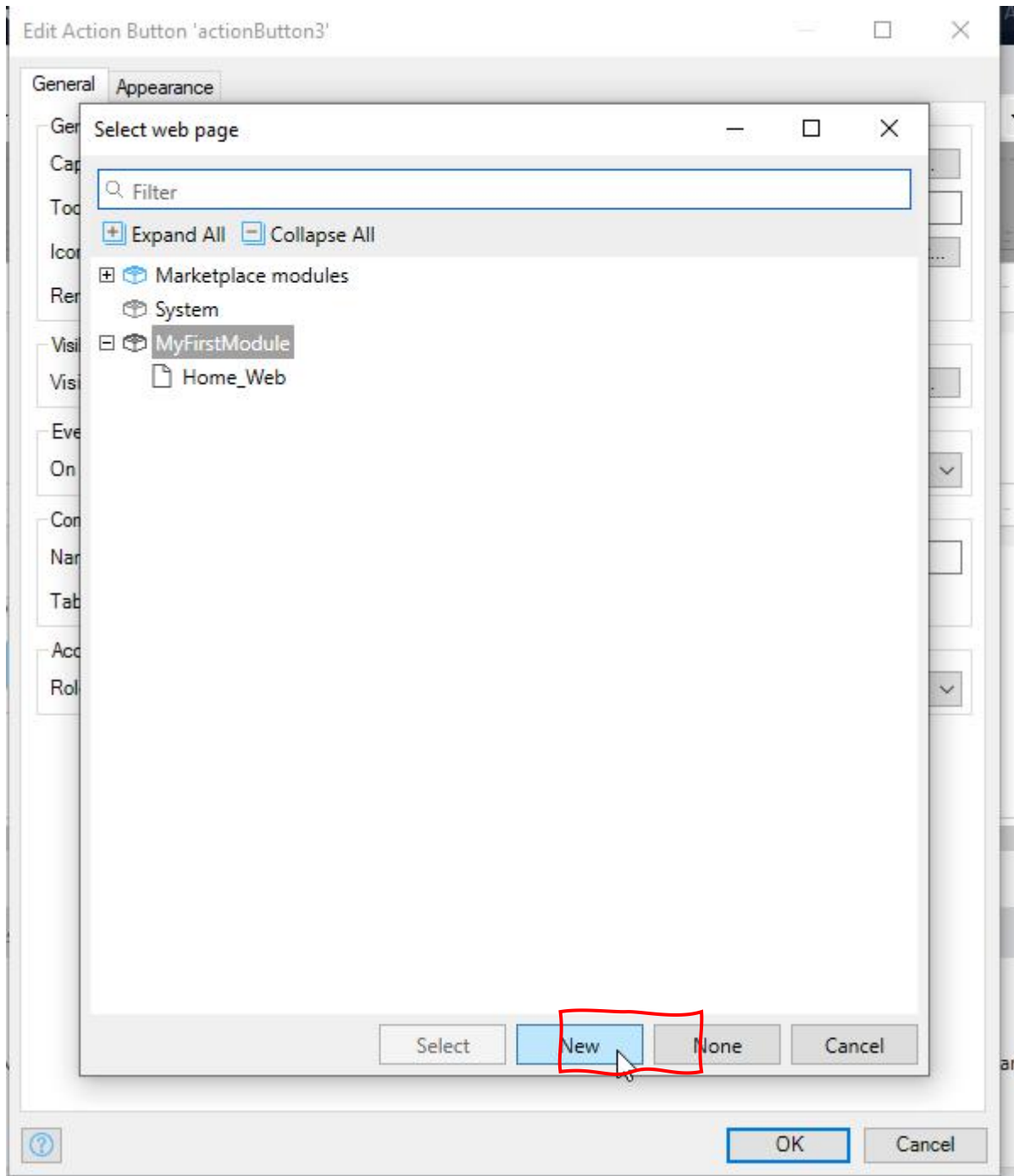


6. 选择好以后，开发工具会自动跳转到第 5 步的页面，给他赋予一个点击事件，我们就让他点击以后跳转一个列

表页面，在 **On click** 下拉列表中选择 **Show a page**



7. 选择好以后系统会弹出一个选择页面的窗体，这时我们还没有其他页面，所以点击 **New** 按钮新创建一个页面。



8. 我们在新弹出的窗口 **Page name** 中填写新页面的名称 Courses_Overview（需要英文），点 book 图标时，我们需要它跳转到一个列表页，选择 Lists->List（默认列表页），在 Navigation layout 下拉框中选择 Atlas_Default（跳转），该选项决定新页面的打开方式，为弹出或跳转等。最后点击 OK 按钮

Create web page

Responsive (Web)

Tablet (Web)

Phone (Web)

Page name

Courses_Overview

Navigation layout

Atlas_Default (Atlas_Core)

Local

Dashboards

Lists

Forms

Detail Pages

Grids

Tabs

Wizards

Workflow

Blank



List



List Columns



List Filtered



List MasterDetail

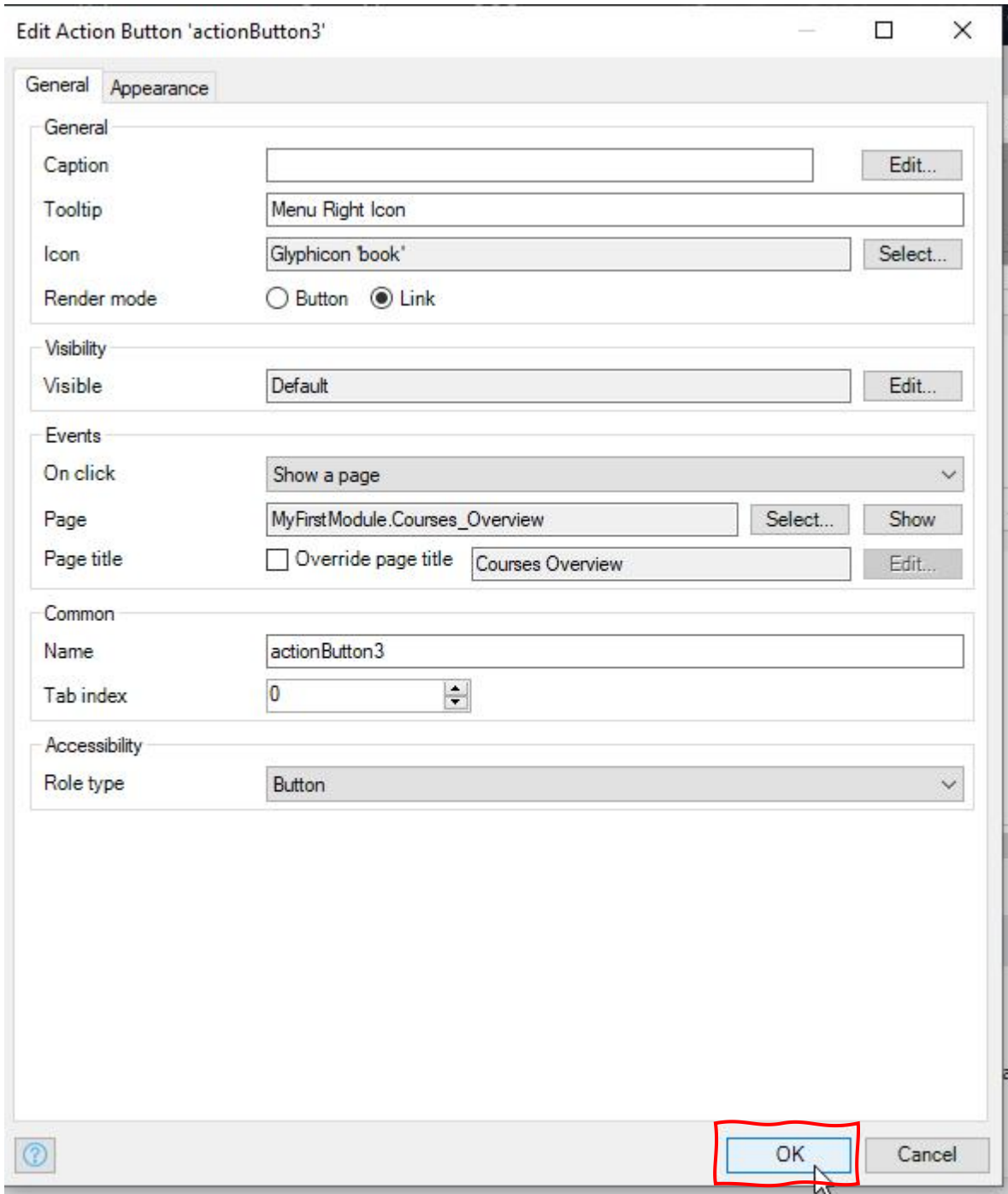


List Status

OK

Cancel

9. 开发工具会自动跳转到第 5 步的图标按钮设置页面，点击 OK 该功能和跳转页面就创建好了。



Dialog Title: Edit Action Button 'actionButton3'

Tabs: General, Appearance

General Section:

- Caption: [Text Field] [Edit...]
- Tooltip: Menu Right Icon
- Icon: Glyphicon 'book' [Select...]
- Render mode: ☐ Button ☒ Link

Visibility Section:

- Visible: Default [Edit...]

Events Section:

- On click: Show a page [v]
- Page: MyFirstModule.Courses_Overview [Select...] [Show]
- Page title: ☐ Override page title Courses Overview [Edit...]

Common Section:

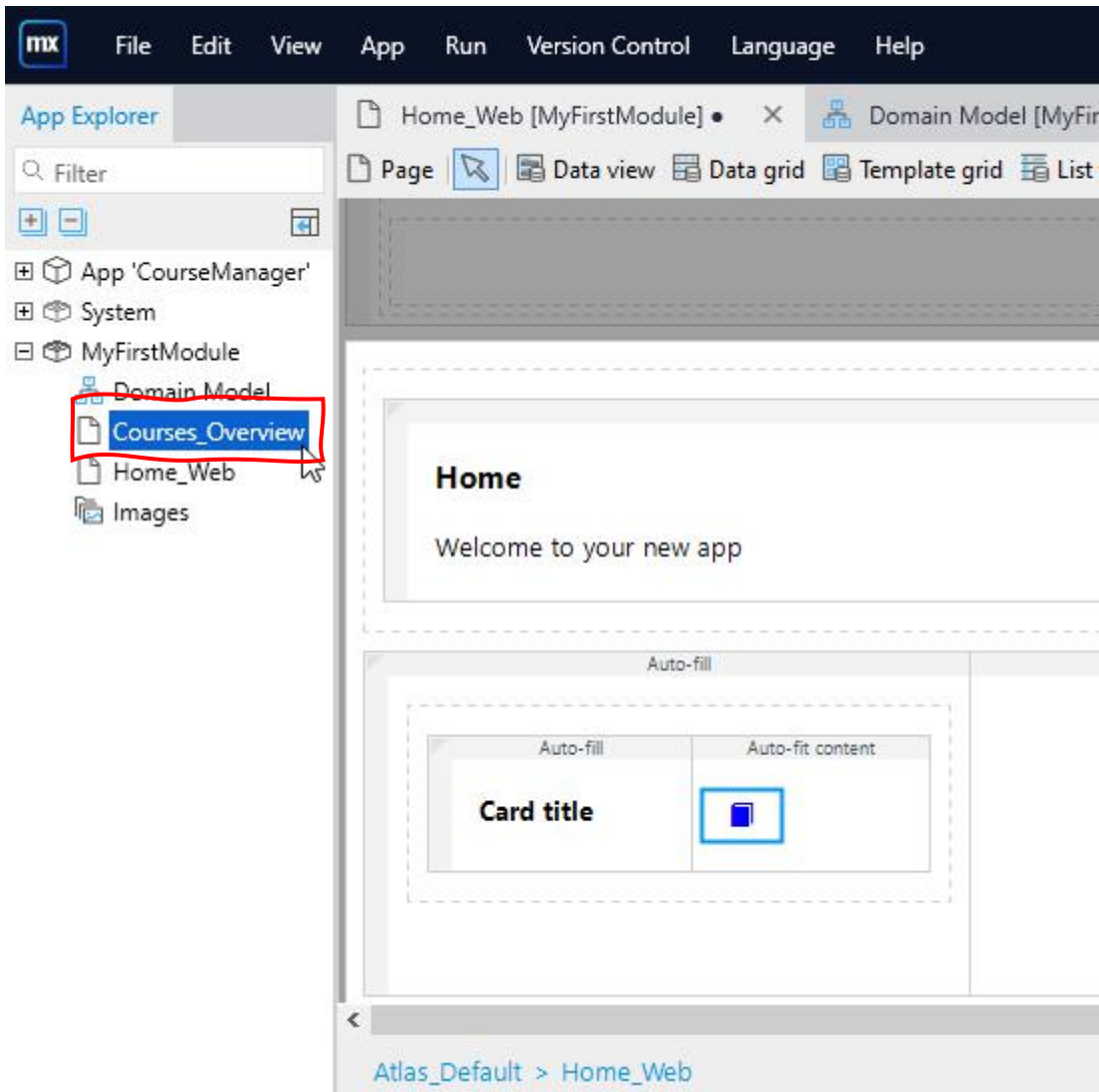
- Name: actionButton3
- Tab index: 0 [Spinner]

Accessibility Section:

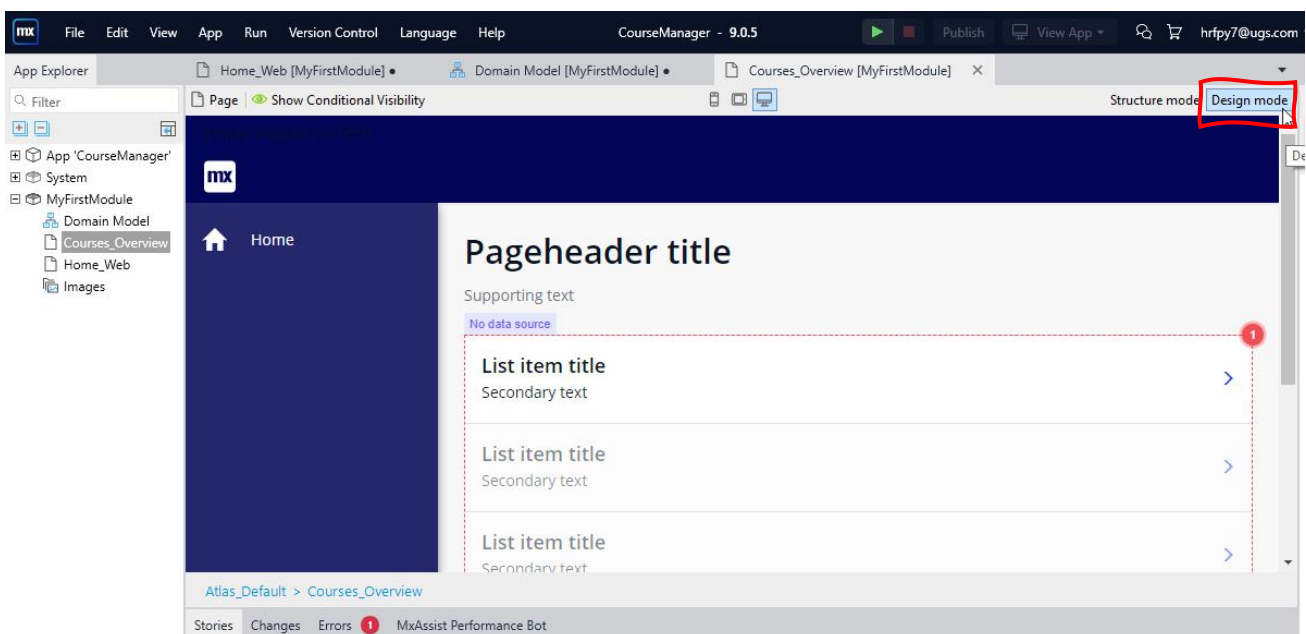
- Role type: Button [v]

Buttons: [?] [OK] [Cancel]

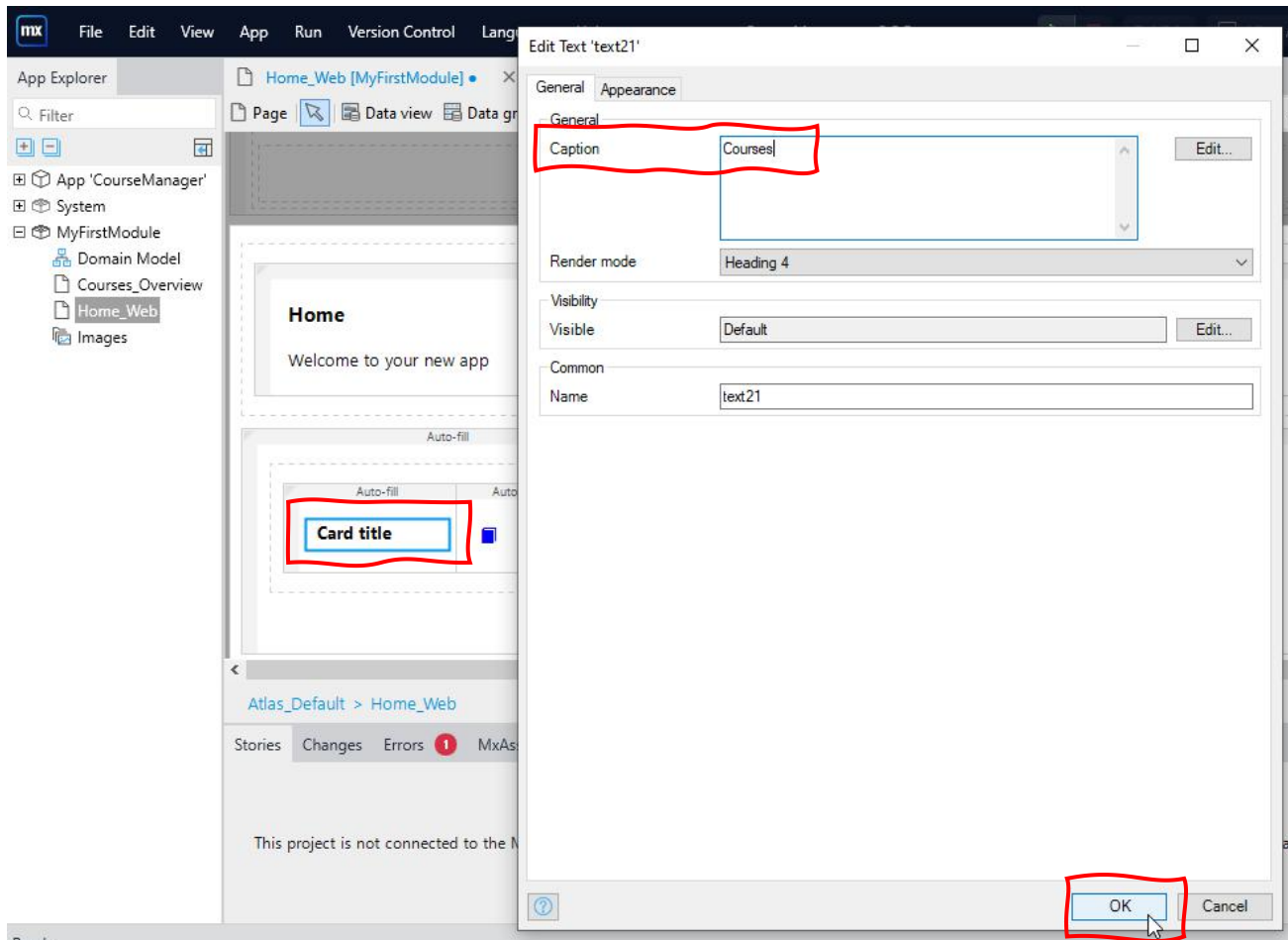
10. 创建完成以后，我们刚才创建的页面就会出现在左 1 的目录树。



刚创建的页面在 Design Mode 展示界面中如下：

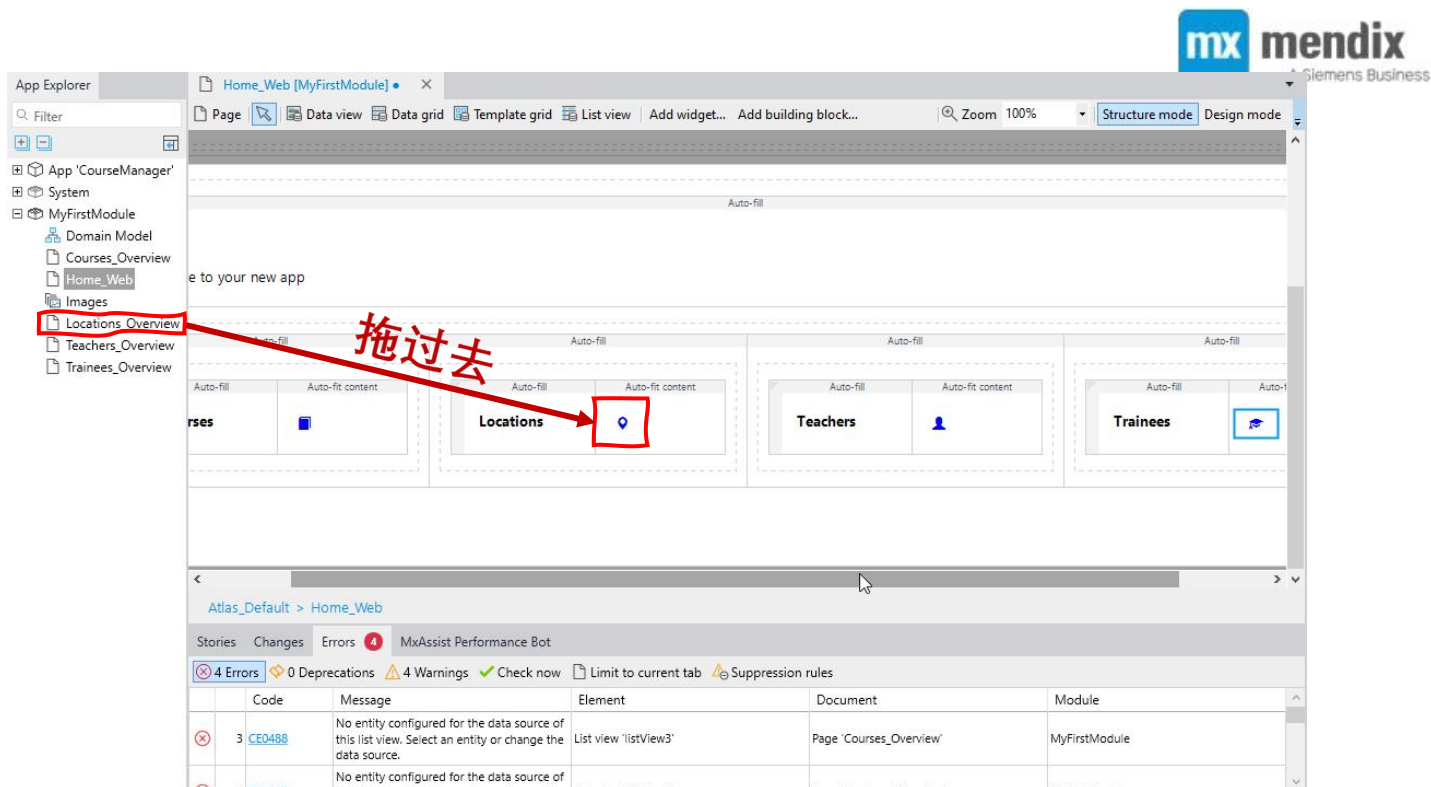


11. 最后双击 **Card Title**，在 **Caption** 输入框中将名字改为 **Courses**，然后点击 ok 就可以了。



12. 其他几列也同样的循环 4-13 步骤，我这里列举一下需要选择图标、列名称、和页面命名。

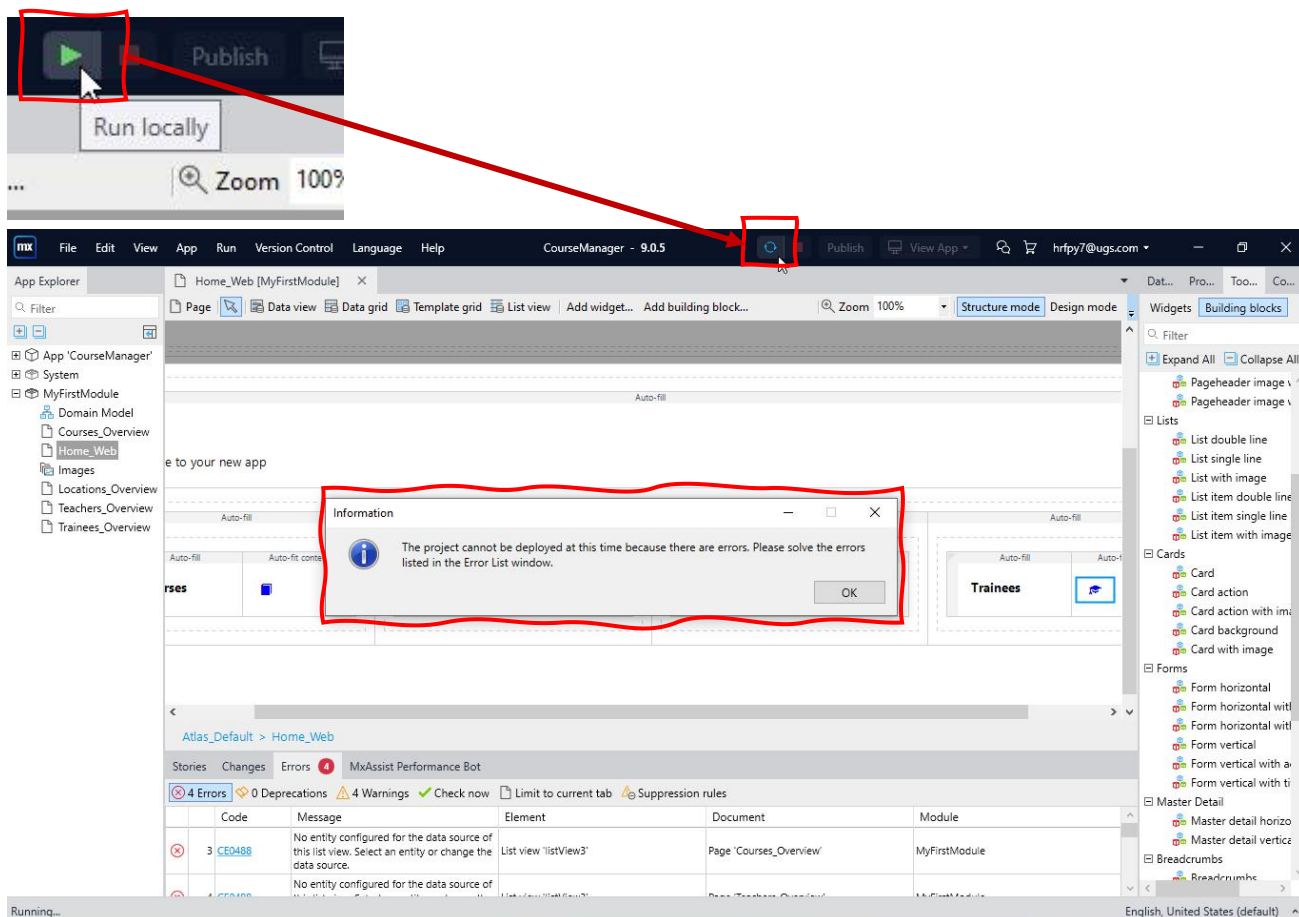
名称	图标	页面命名
Courses	book	Courses_Overview
Locations	map	Locations_Overview
Teachers	user	Teachers_Overview
Trainees	education	Trainees_Overview



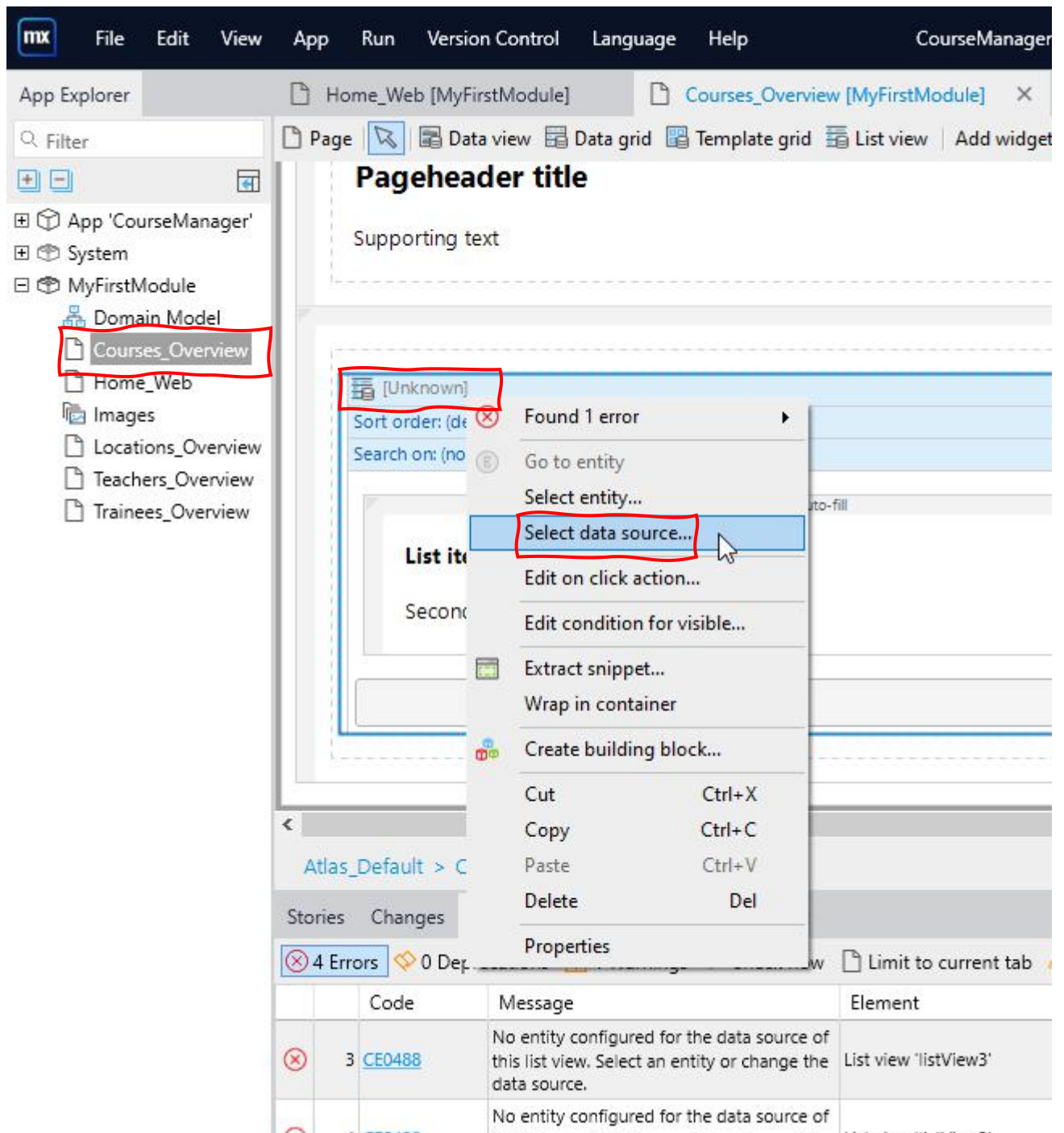
(一个捷径：右击已创建的 Overview 页面，选择 Duplicate 复制，右键 Rename 成新的 Overview 页面，将新的页面拖动至对应的图标上，可以快速设置跳转至该页面。)

4. 数据源绑定

1. 现在，点击上方 Run Locally 按钮，会发现运行报错，因为我们还没有跟数据进行绑定。

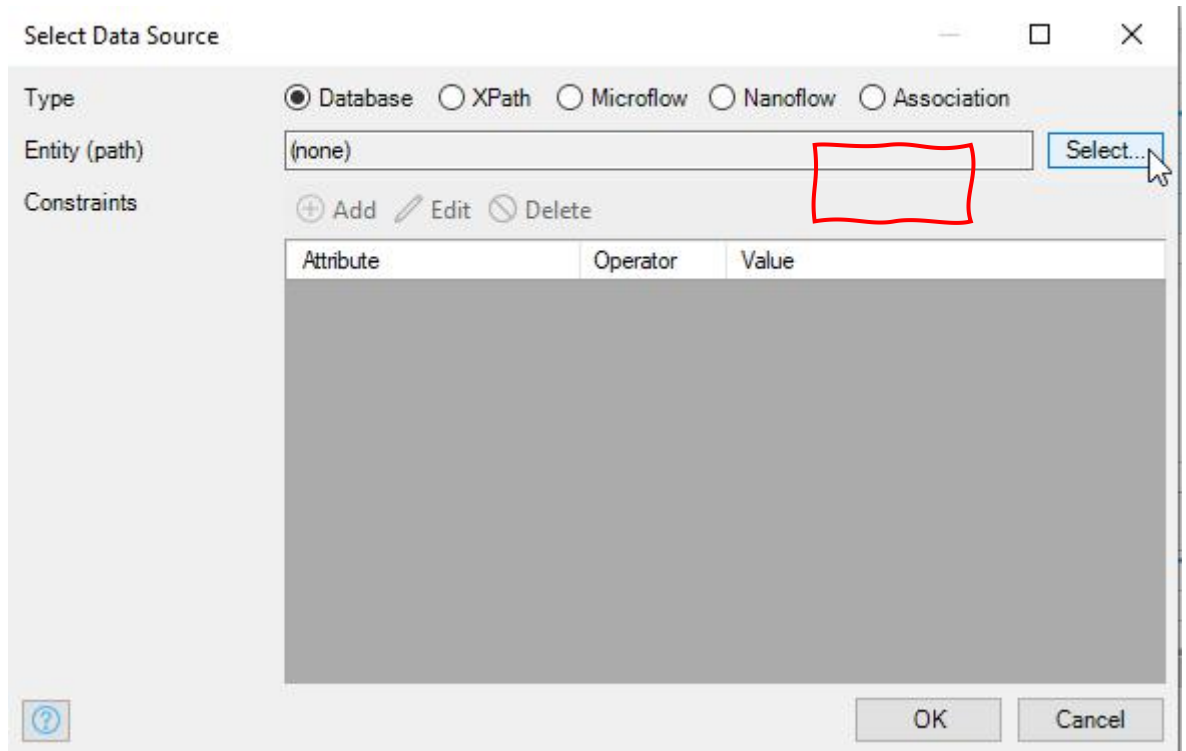


2. 将页面跟数据进行绑定，双击 **Courses_Overview** 页面，在 **Unknown** 右击选择 **Select data source** (选择数

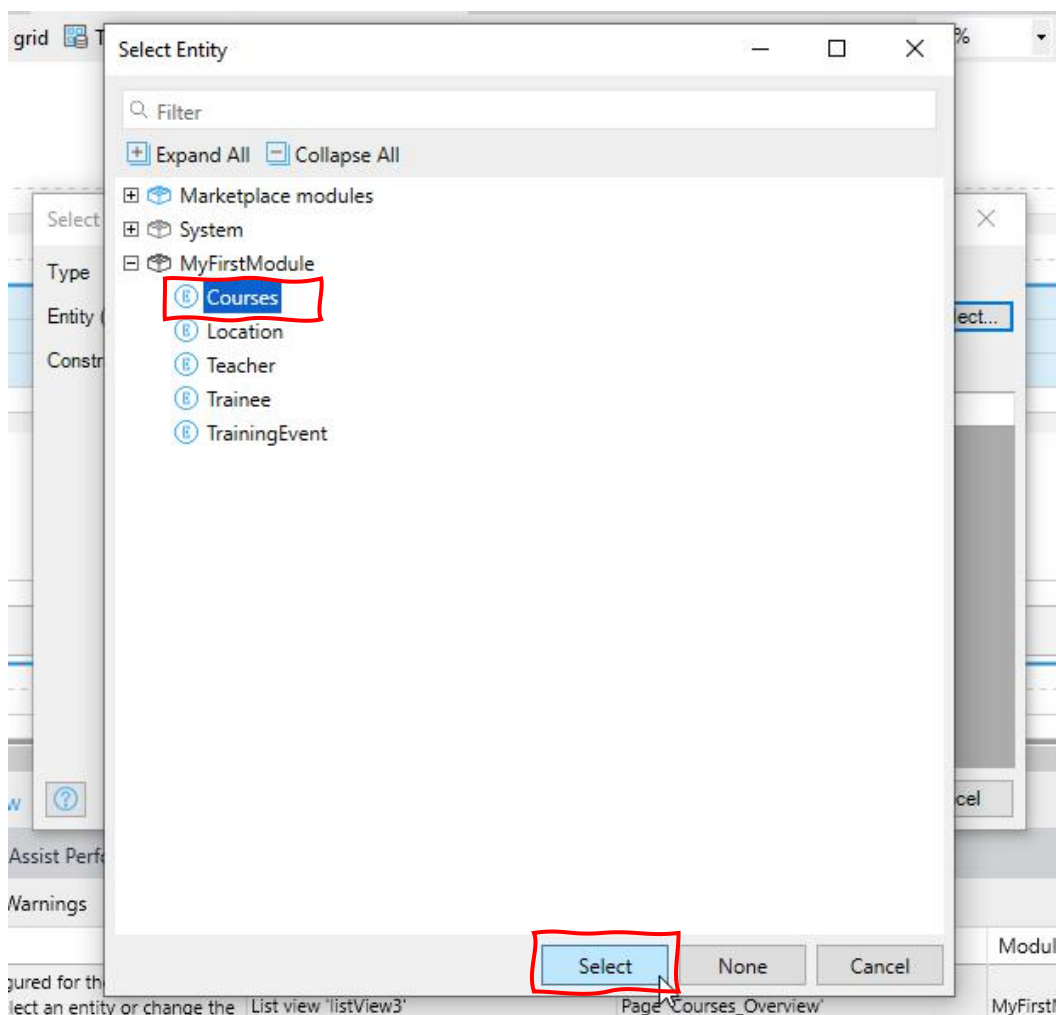


据源)

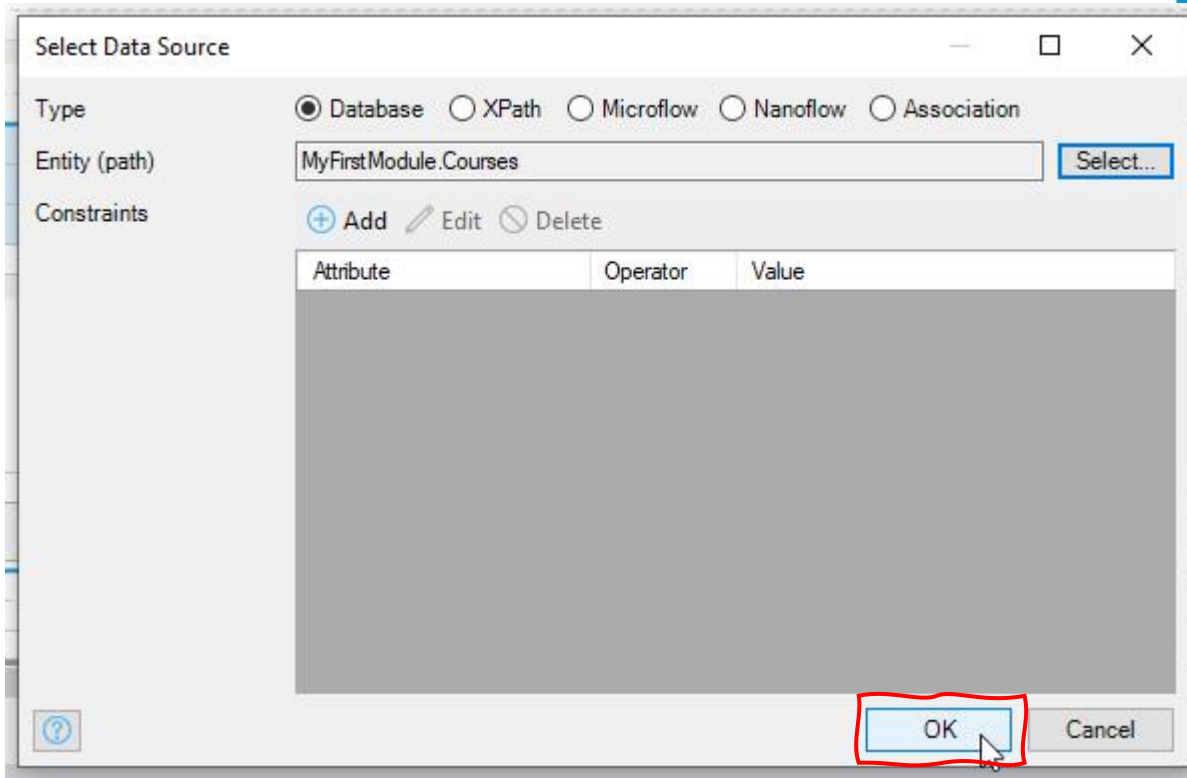
3. 在弹出的页面上，Entity (path) 处 点击 **Select** 按钮，会弹出选择 Entity 的页面。选择数据模型 Courses，点



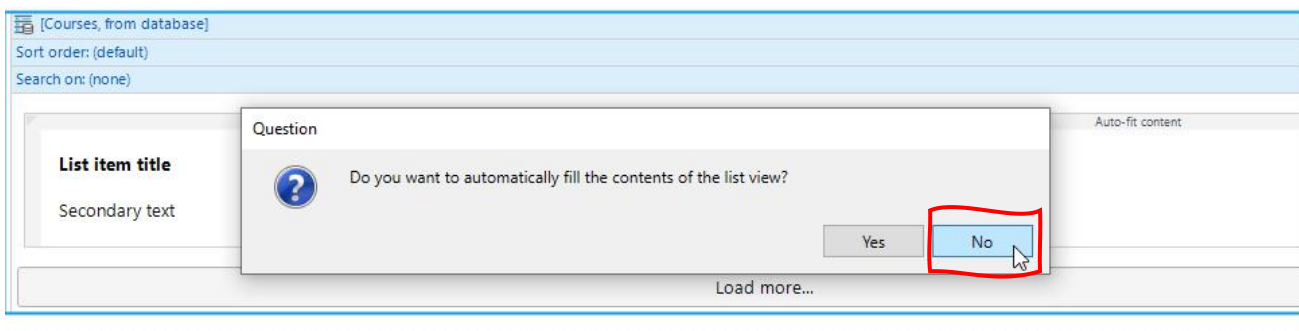
击 Select 按钮。



4. 开发工具会自动跳转到 3 步骤，点击 OK 按钮，该页面就和 Courses 数据模型绑定成功。

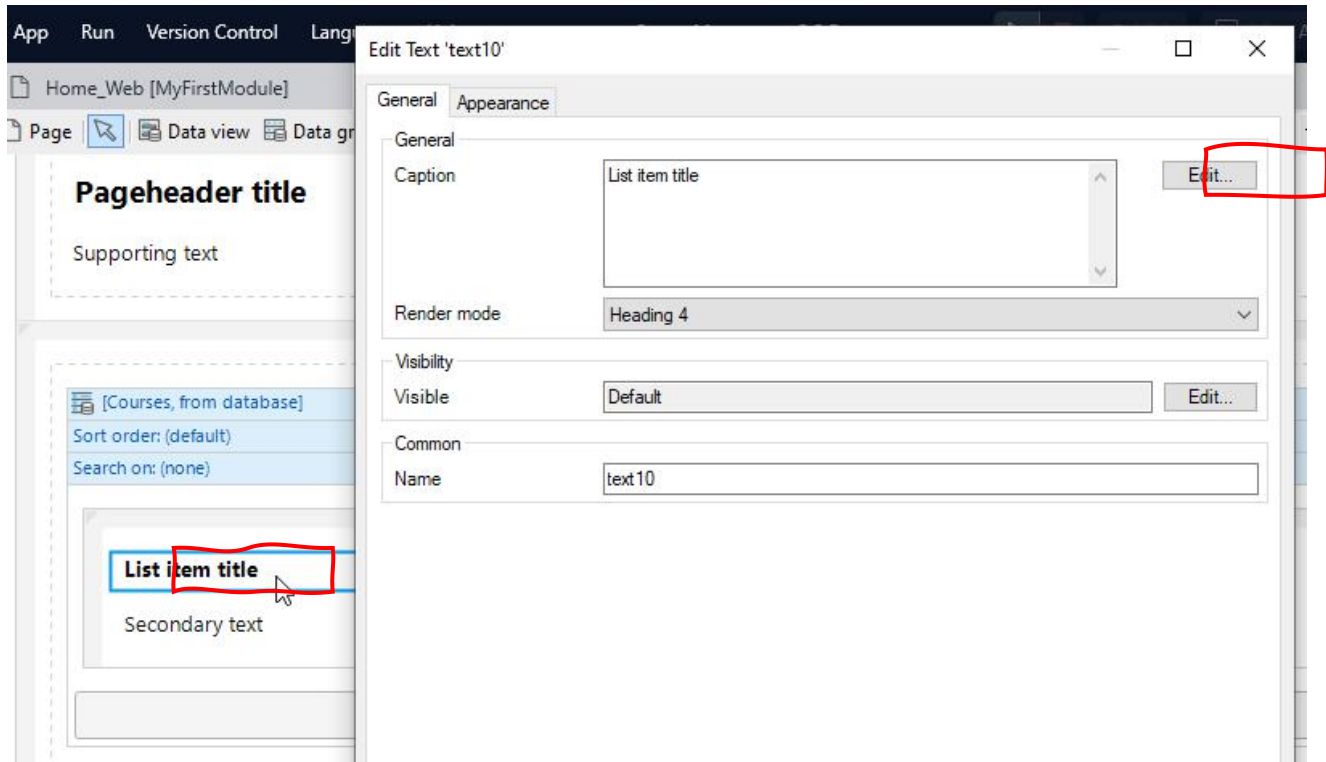


5. 开发工具会弹出一个提示框来询问我们是否自动的响应页面属性，我们选择 NO 按钮不需要

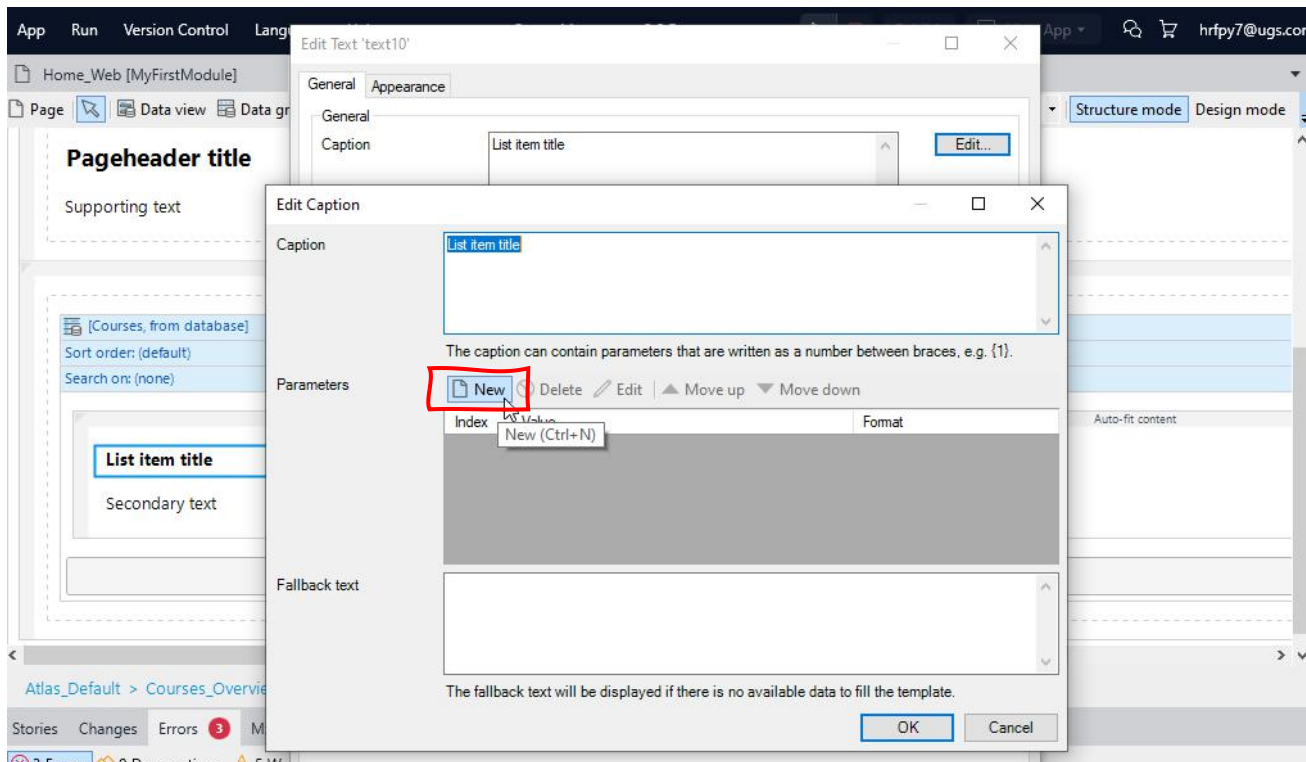


5. 属性绑定

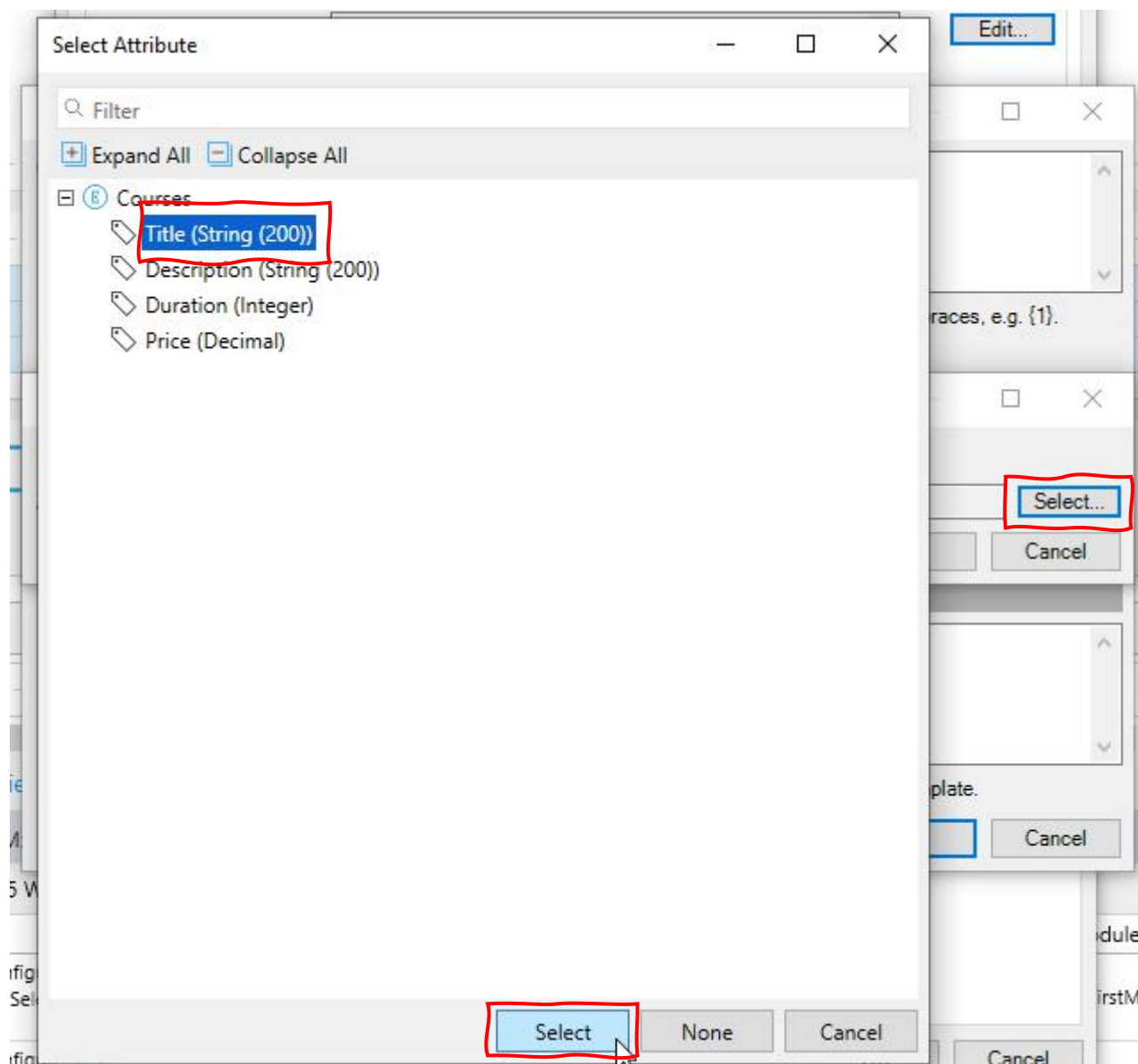
1. 双击 页面上的 **List item title** 进行属性绑定，在弹出的窗口选择 **Edit** 按钮，会弹出 Edit Caption 的窗口。



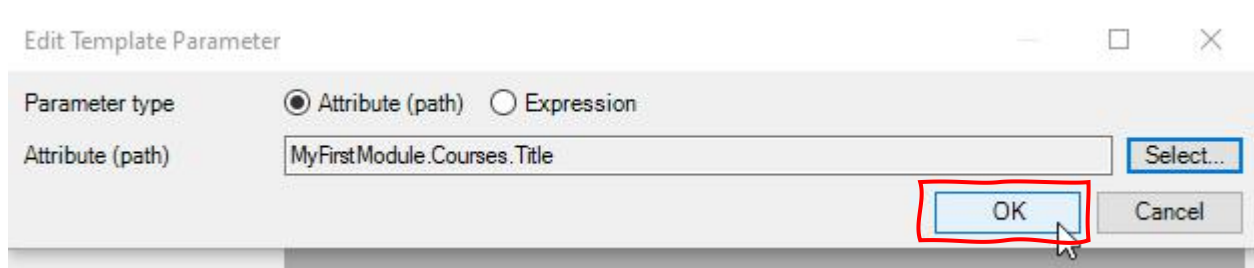
2. 在该窗口选择 New 按钮，会弹出 Edit Template Parameter 编辑参数的窗口。



3. 在 Edit Template Parameter 编辑参数的窗口中 **Attribute (Path)** 边上选择 **Select**，绑定 Attribute 属性：开发工具会弹出 Courses 数据模型的所有属性，选择 Title 属性，再点击 Select 按钮。



4. 开发工具会回到之前 Edit Template Parameter 编辑参数界面，点击 OK 按钮，完成选择。



5. 这是系统会**自动**出现一个 Index 序号, 我们将这个需要输入到 Caption 中 (页面对应位置会显示{1}代表的的数据), 点击两个 OK 按钮->OK 按钮, 完成数据绑定。

Edit Caption

Caption

{1}

The caption can contain parameters that are written as a number between braces, e.g. {1}.

Parameters

New Delete Edit Move up Move down

Index	Value	Format
{1}	Title	

Fallback text

The fallback text will be displayed if there is no available data to fill the template.

OK Cancel

6. 用相同的方法，将其他 **属性** 进行绑定。

Edit Caption

Caption

{1}
{2}天
¥ {3}

Parameters

New Delete Edit Move up Move down

Index	Value	Format
{1}	Description	
{2}	Duration	123456789
{3}	Price	123456789.00

Fallback text

The fallback text will be displayed if there is no available data to fill the template.

OK Cancel

[Courses, from database]

Sort order: (default)

Search on: (none)

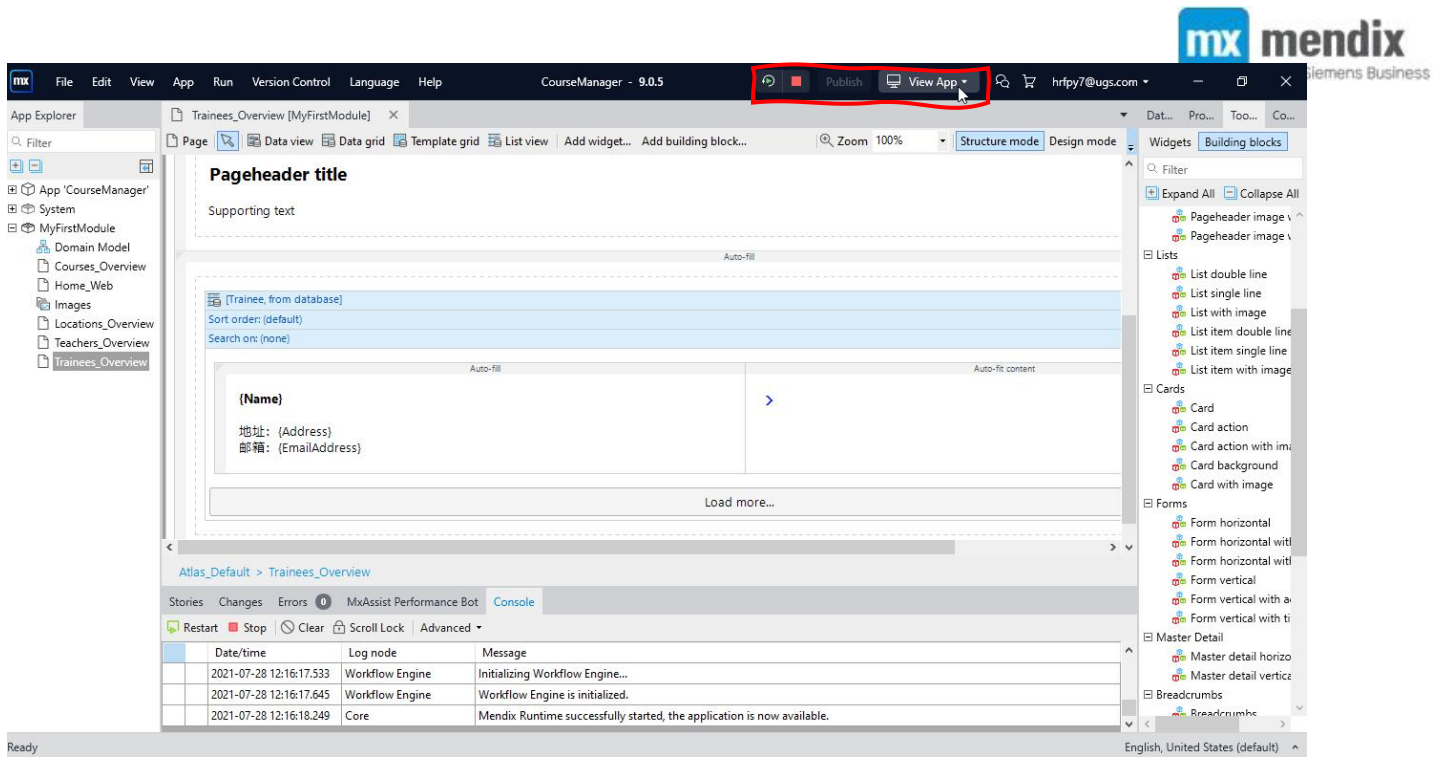
Auto-fill

{Title}

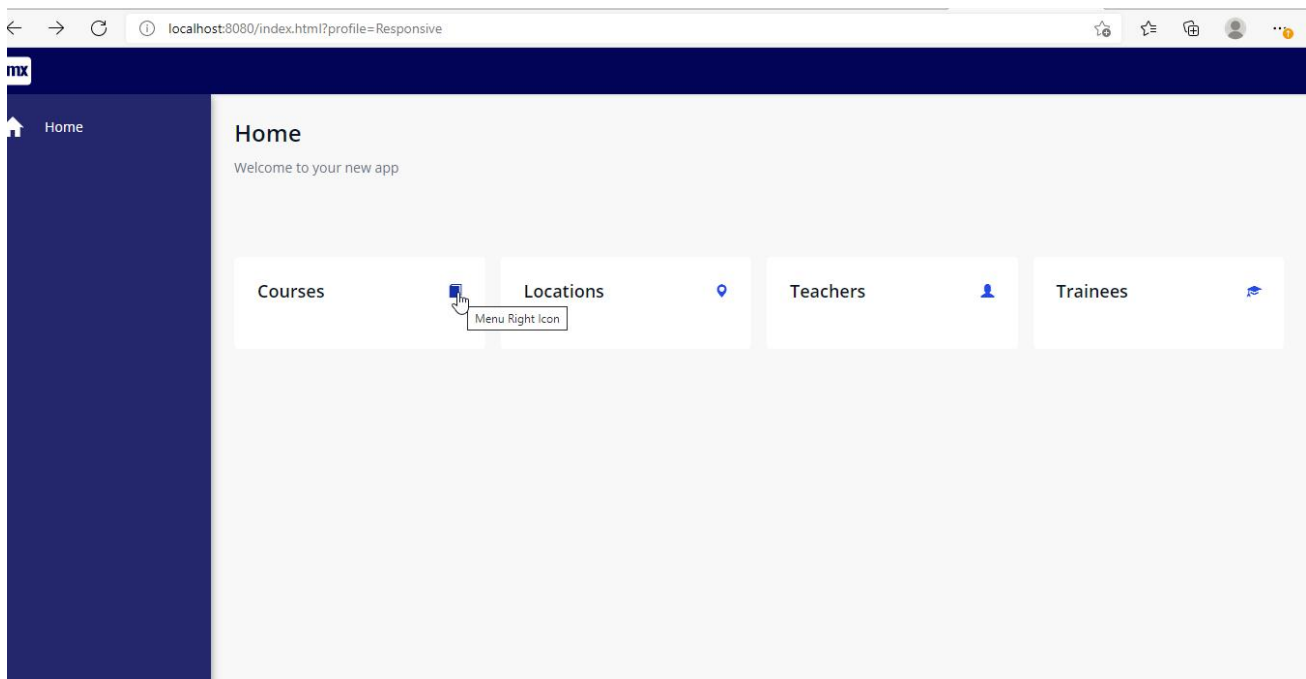
{Description}
{Duration}天
¥ {Price}

Load more...

7. 根据 [4.数据源绑定](#) 和 [5.属性绑定](#) 的步骤，将其余页面 (**Locations_Overview**、**Teachers_Overview**、**Trainess_Overview**)进行数据绑定。然后我们点击 Run Locally 运行项目，然后点击 View 按钮进行预览。

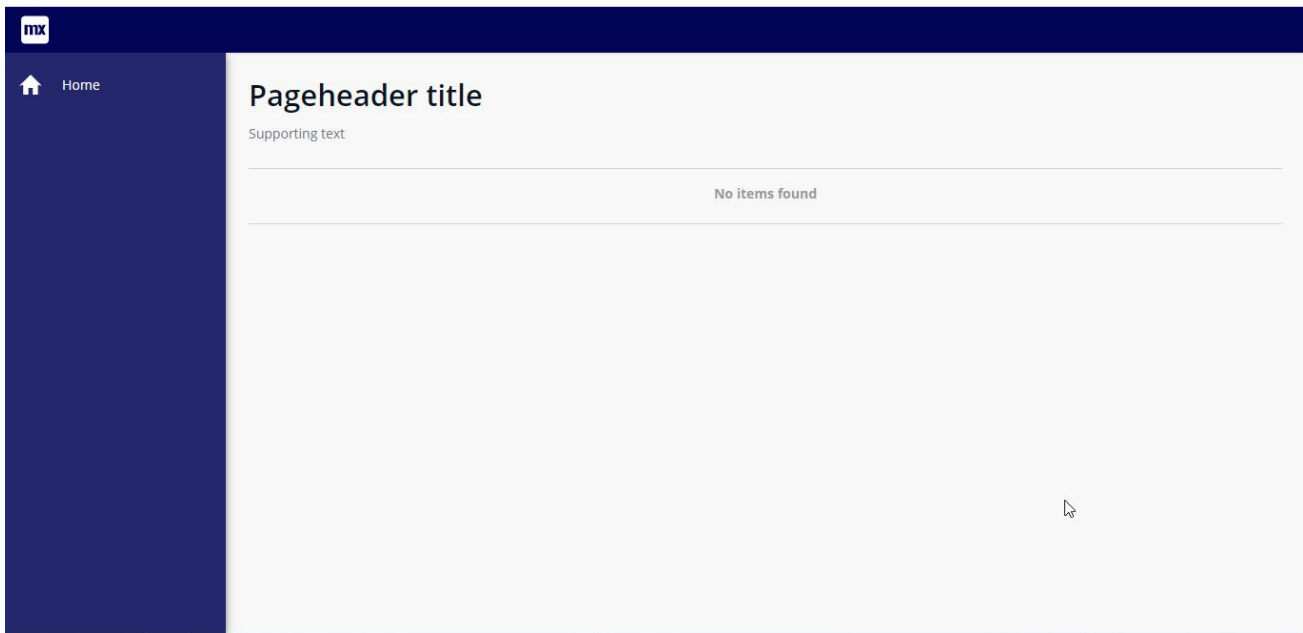


8. 我们可以点击我们创建的四部分，进行查看。

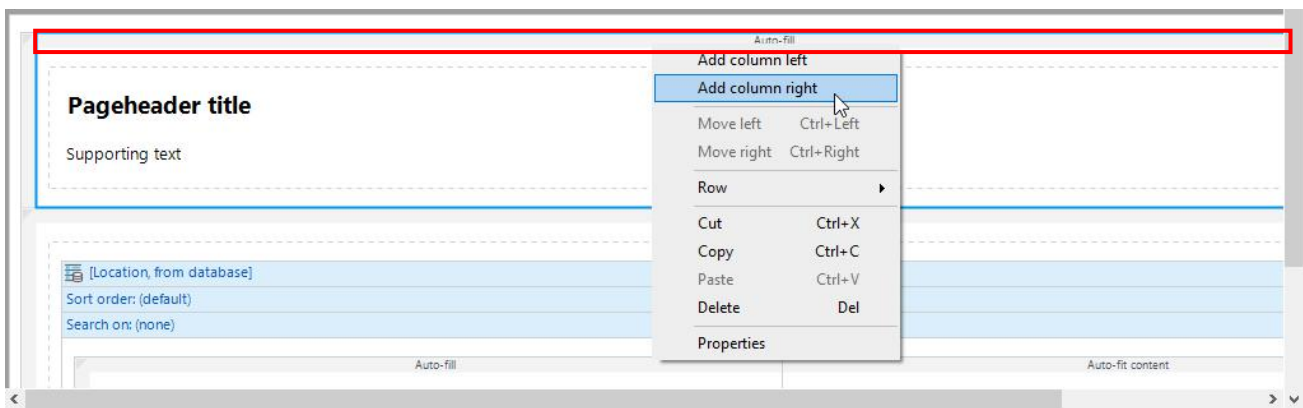


6. 数据新增

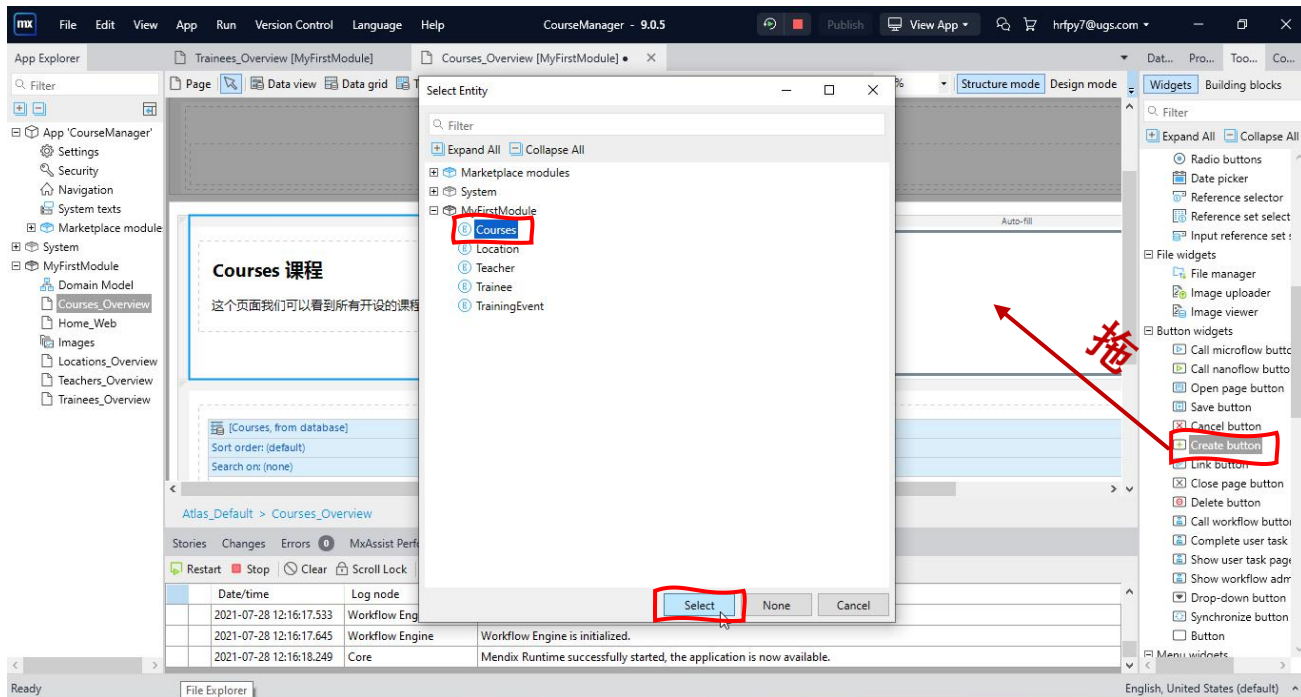
1. 我们可以发现，里边现在是空的，没有数据的。所以我们现在需要实现新增功能。



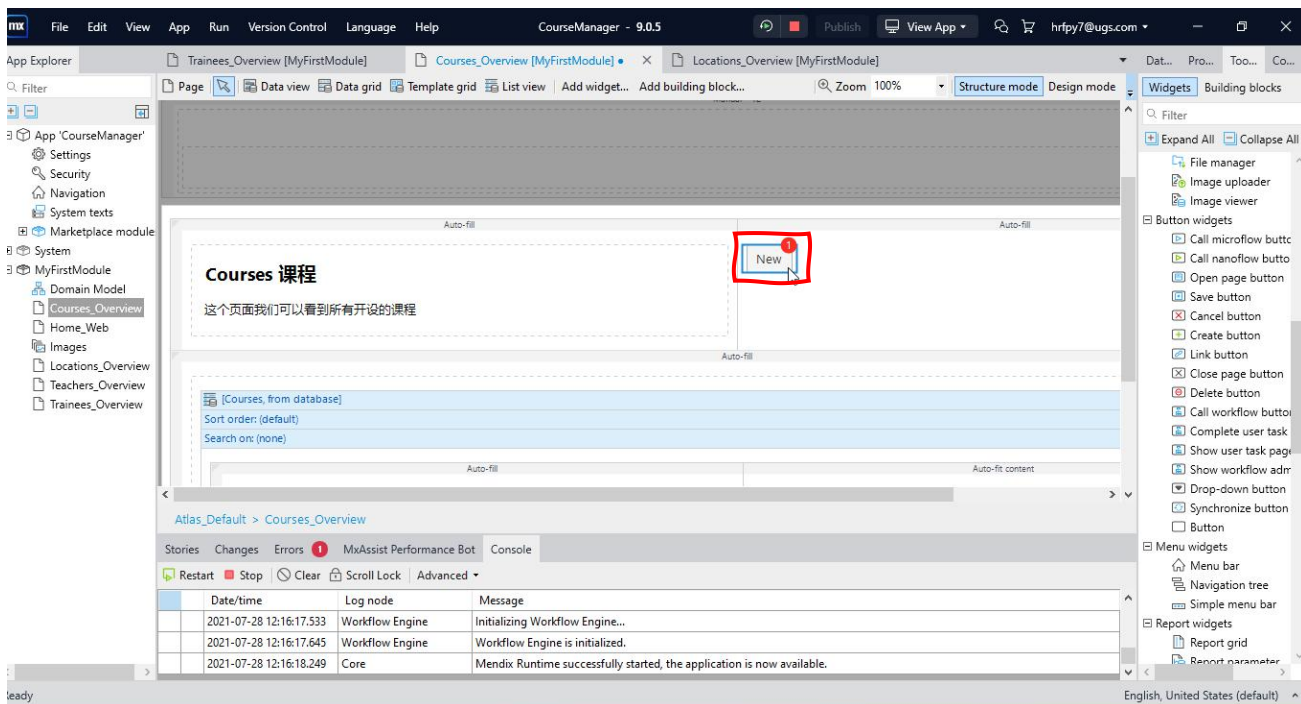
2. 以 Courses_Overview 列表页面为例，实现**新增课程（add courses）**功能。双击目录树 **Courses_Overview** 页面，新增一个新的列 add column left/right, 可自行修改 Pageheader title 页面标题为 Courses 课程和 Supporting text 为描述内容。



- 从 **3 右** Widgets 中找到 Buttons，将 Create Button 拖动到右边的那一列中，将弹出 Select Entity 界面（选择创建的实体）。选择 Courses，点击 Select，界面消失。



- 双击带有 New 字样的 button 按钮，弹出编辑“新建”按钮的页面。



5. 我们可以通过清空 caption 栏，Icon 栏选择 plus，Button Style 选择 success，更改按钮式样。更多式样也可以自行探索一下。
6. 紧接着我们在选择对应的页面，点击 On click page 中的 Select 按钮，来选择对应的新增页面。

Edit Action Button 'actionButton1'

General Appearance

General

Caption

Tooltip

Icon

Render mode

Button style

Visibility

Visible

Events

On click

Entity (path)

On click page

Page title

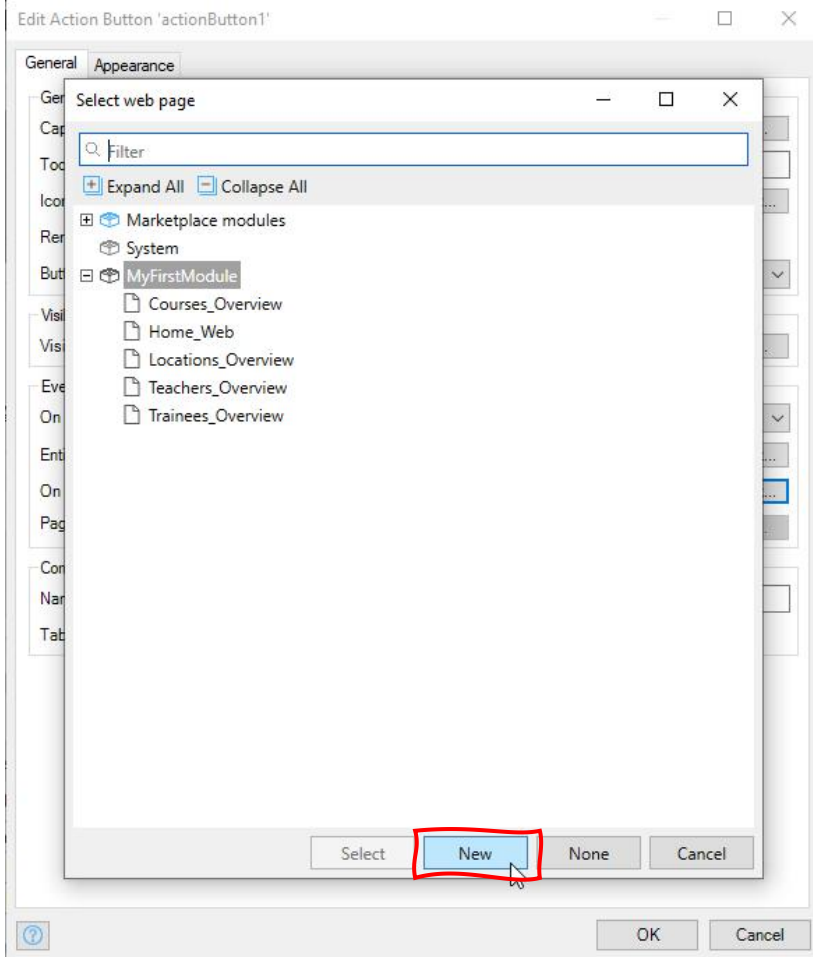
Common

Name

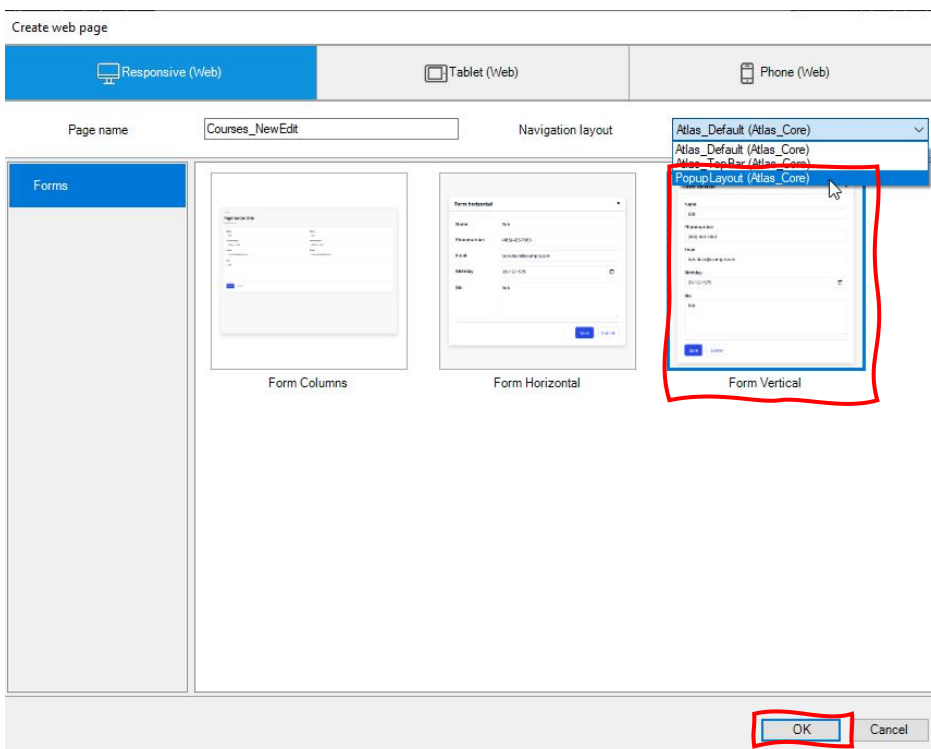
Tab index

OK Cancel

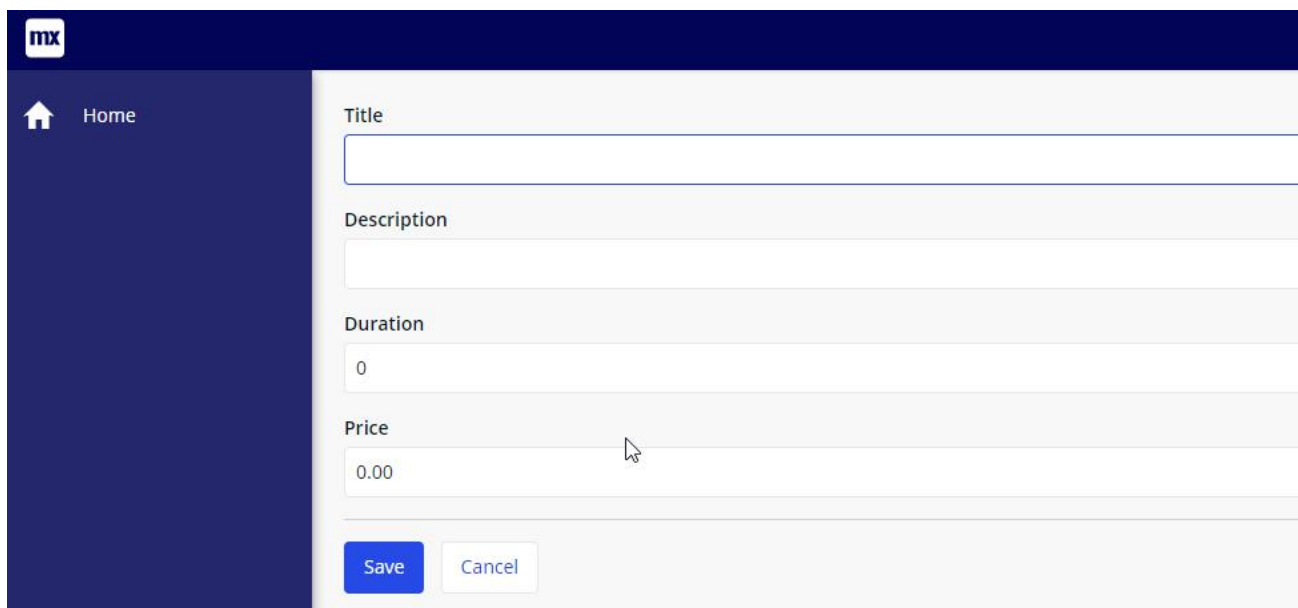
7. 这个时候我们可以发现，我们是没有页面可以新增课程的，所以点击 New 按钮，创建一个新增页面。



8. **Page name** 会自动为命名，不需要更改。在 **Navigation layout** 下拉选项中选择 **PopupLayout**（弹出页面方式），然后选择 **Forms** 中的 **Form Vertical**。然后点击 **Ok->Ok** 按钮，完成页面创建。



9. 完成并保存以上操作以后，我们**重新点击 Run Locally** 运行我们的项目，浏览一下。点击 **Courses** 菜单进入列表页，在列表页上点击右上角的 **Add** 按钮，这时候我们就可以发现，有弹出新增页面，可以新增数据了。

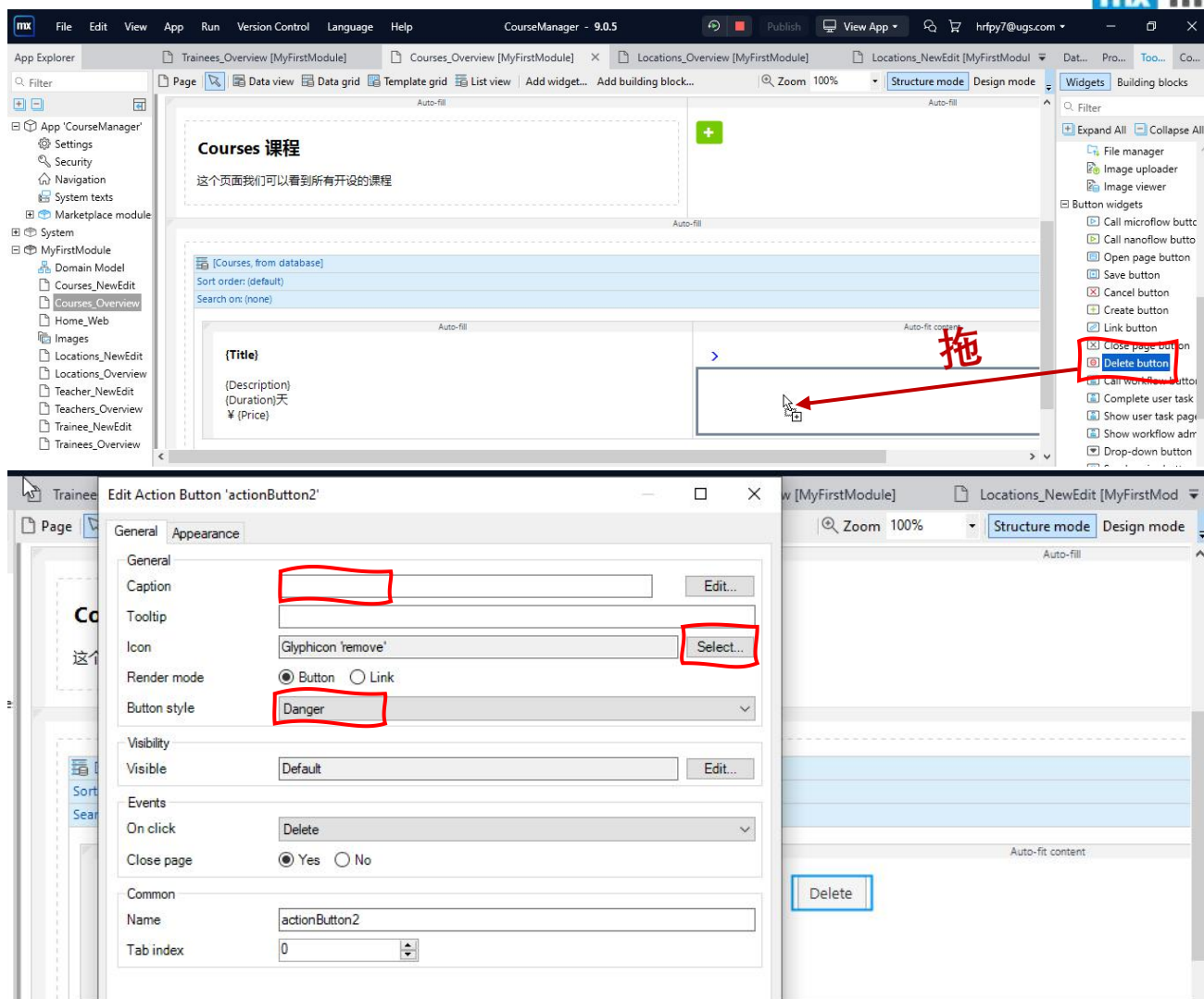


10. 重复[当前章节](#)的操作步骤，完成对其他 (Locations_Overview、Teachers_Overview、Trainess_Overview)页面的新增功能。新增一列，拖一个 create button，选择对应的 Entity，创建新增页面。完成以上操作以后，所有的列表页面都有新增功能了。

7. 数据删除

以 Courses 为例，双击打开 Courses_Overview 页面，在 **3 右** **ToolBox** 中的 **widgets** 中搜索 **button** 按钮，找到 **Delete button**，拖入到每一列详细信息栏的右边栏，就完成删除的操作了。重复当前步骤完成其他 (Locations_Overview、Teachers_Overview、Trainess_Overview)页面的删除功能。

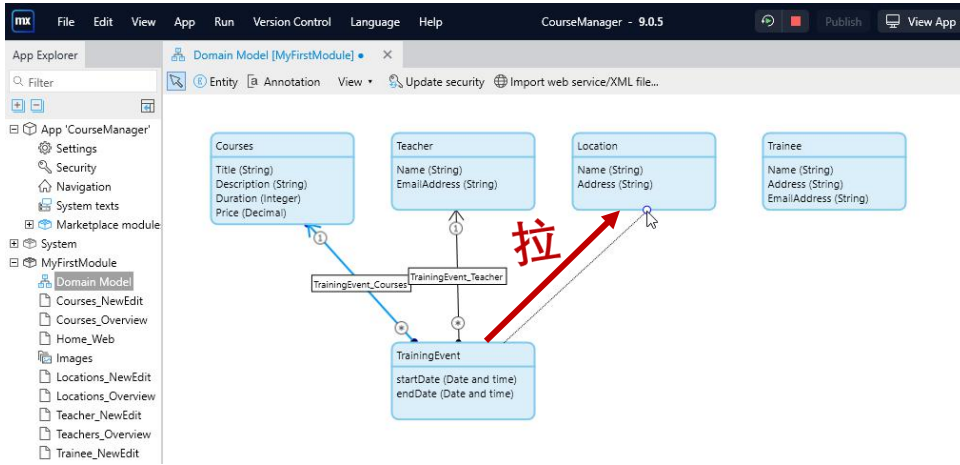
下图包含了使显示红色✖按钮的样式参数。



8. 关系型数据模型

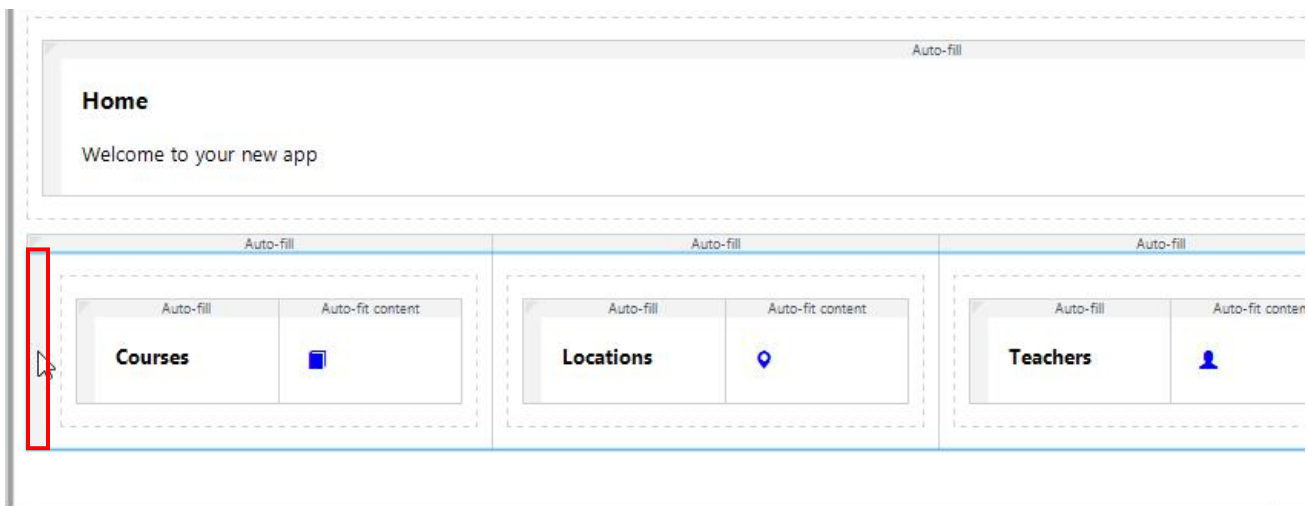
1. 经过以上操作我们就大体实现了四大模块（课程、老师、地址、学员）的新增和删除工作了。现在根据学校的实际需求出发，学校的维护人员需要发布培训活动，培训活动需要包含课程和培训课程的老师，还有培训地址，方便学员进行报名。需求分析完毕，接下来我们开始操作。

我们首先需要在数据模型中将 **TrainingEvent**（培训活动）的模型与课程、老师、地址进行一对多的关联，方便维护人员进行维护。双击 [1] 左目录树中的 **Domain Model**，然后在打开的窗口中从 **TrainingEvent** 模型向其他三个模型（**Courses**、**Teacher**、**Location**）拉箭头，完成模型一对多的绑定。

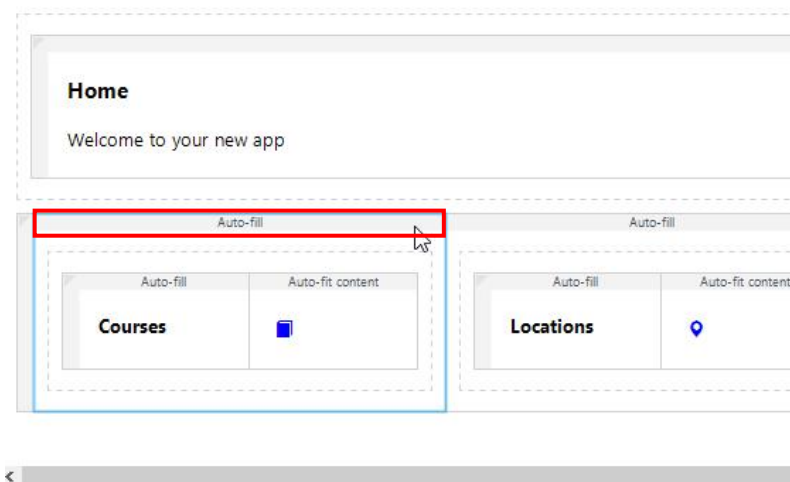


2. 数据模型绑定成功以后，我们需要在 **Home_Web** 主页面添加新模块的入口，首先介绍一下在 **MendixPro** 中如何操作表格。

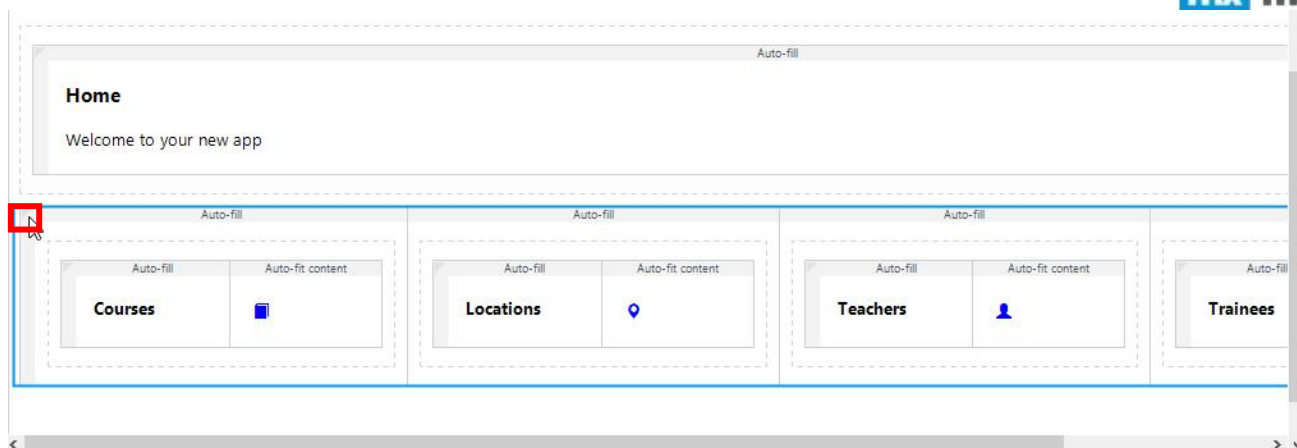
如何选择行：鼠标点击下图所示区域是选择一整行



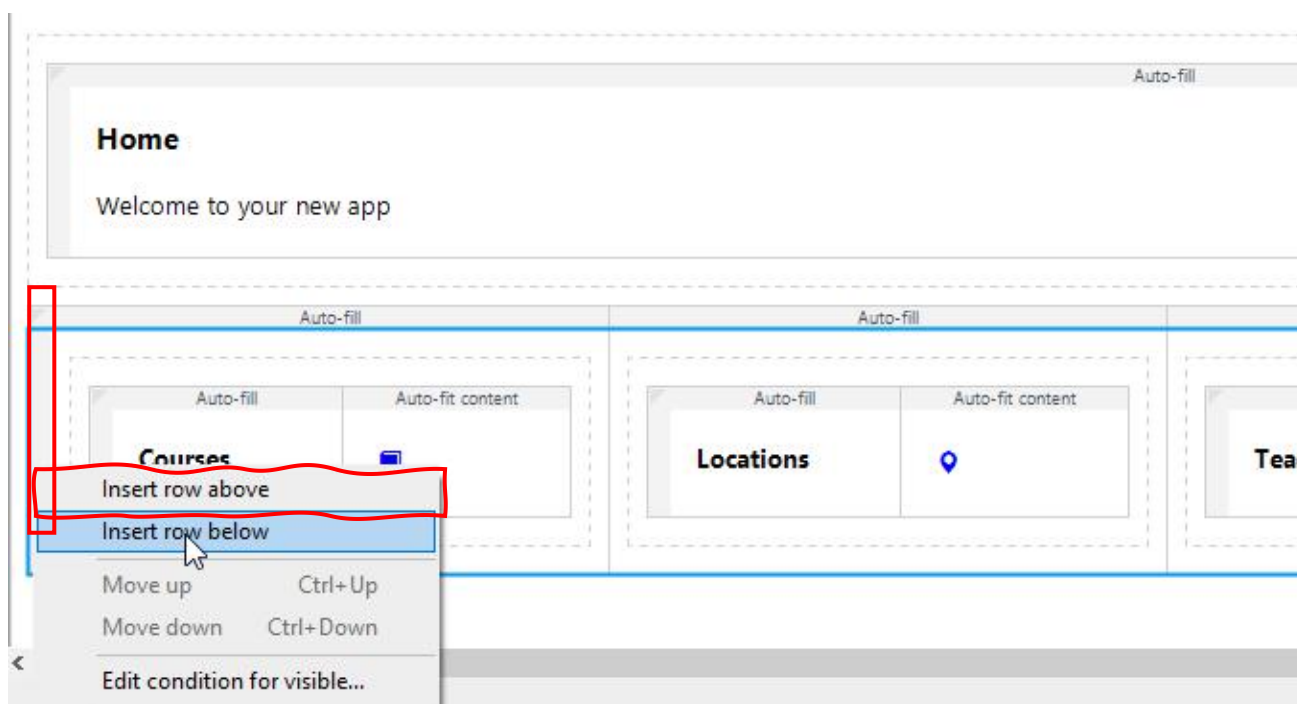
如何选择列：鼠标点击下图所示区域是选择对应的列



如何选择表格：鼠标点击下图所示区域是选择整个表格



3. 双击 **1** 左目录树中的 Home_Web 页面，选择行以后鼠标右击选择 Insert row below。



4. 选择 Insert row below 以后，页面会出现让我们选择将新建的行分成几等份，我们直接跟上一行一样，分成四等份即可。
5. 根据下列名称、图标、及页面命名，完成 [3.创建页面](#)（3-13），[4.数据源绑定](#)，[5.属性绑定](#)，[6.数据新增](#)的内容。

名称	图标	页面命名
TrainingEvent	calendar	TrainingEvent_Overview

Training Event 培训活动

这个页面我们可以看到所有发布的培训活动

Auto-fill

[TrainingEvent, from database]

Sort order: (default)

Search on: (none)

Auto-fill

{.../Title}

{startDate} ~ {endDate}

By {.../Name} 【联系方式: {.../EmailAddress}】

At {.../Name} -- {.../Address}

Atlas_Default > TrainingEvent_Overview

6. 完成以上操作以后，我们就可以对培训活动进行新增操作，我们可以看到在新增页面会自动带出课程和培训课程的老师，还有培训地址信息，到这一步就完成了关系型一对多数据模型的创建（可以在 TrainingEvent_NewEdit 中把 Courses 拖到最上面，让表格更合理。）

mx Home

Training

这个页面我们可以看

Auto-fill

start date

m/d/yyyy

end date

m/d/yyyy

Courses

Teacher

Location

Save Cancel

Auto-fill

[TrainingEvent, page parameter]

Courses

[TrainingEvent_Courses/Courses/Title]

start date

[startDate]

end date

[endDate]

Teacher

[TrainingEvent_Teacher/Teacher/Name]

Location

[TrainingEvent_Location/Location/Name]

Mendix 敏捷开发零基础学习《二》-进阶

Microflow、表单验证、运算符、条件判断、数据嵌套、触发器、Debug 问题跟踪、版本管理

- 该章节的例子是基于 [【Mendix 敏捷开发零基础学习《一》】](#) 的需求和例子来讲解。

如果是第一次尝试，最佳跟练顺序为：[敏捷开发《一》](#) → [一、微流【3.1 举例 1】](#) → [三、数据嵌套](#) → [一、微流【3.2 举例 2】](#) → [二、Debug](#) → [一、微流【3.2.2 结束日期统计\(14~21\)】](#) → [一、微流【3.3 举例 3】](#) → [敏捷开发《三》](#)

目录

一、Microflow 微流

- 1、概述
- 2、常见的功能
- 3、微流可以做那些事情?
 - 3.2.1 报名数量统计
 - 3.2.2 结束日期统计
 - 3.3.1 业务需求
 - 3.3.2 需求分析
 - 3.3.3 项目实施

二、DeBug 问题跟踪

- 1、工具介绍
- 2、问题查找

三、数据嵌套

- 1、需求分析
- 2、项目实施

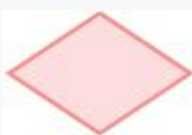
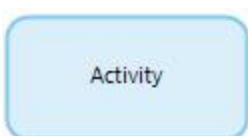
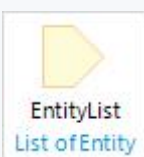
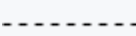
四、版本管理

一、Microflow 微流

1、概述

Mendix 的 **Microflow** 微流其实是我们处理程序逻辑的一种方式。微流可以执行诸如创建和更新对象，显示页面以及做出选择之类的动作。这是表达传统上以文本程序代码结尾的可视化方式。

2、常见的功能

图形	名称	描述
	开始节点	开始事件是微流的起点。一个微流只能有一个启动事件。
	结束节点	结束事件定义微流将停止的位置。在某些情况下，取决于微流的返回类型，必须指定一个值。可以有多个结束事件。
	判断节点	决策根据条件做出决策，并且仅遵循一个流出流。微流中没有并行执行。
	合并节点	合并可用于将多个序列流合并为一个。如果在微流中做出选择，然后需要完成一些常见的工作，则可以使用合并将两个（或更多）路径合并。
	任务节点	任务节点是在微流中需要处理的任務
	参数节点	在触发微流的位置填充参数
	顺序流	顺序流是将事件，活动，决策和彼此合并的箭头。它们共同定义了微流程中的执行顺序。
	注释流程	关联是可用于将注释连接到另一个元素的连接。

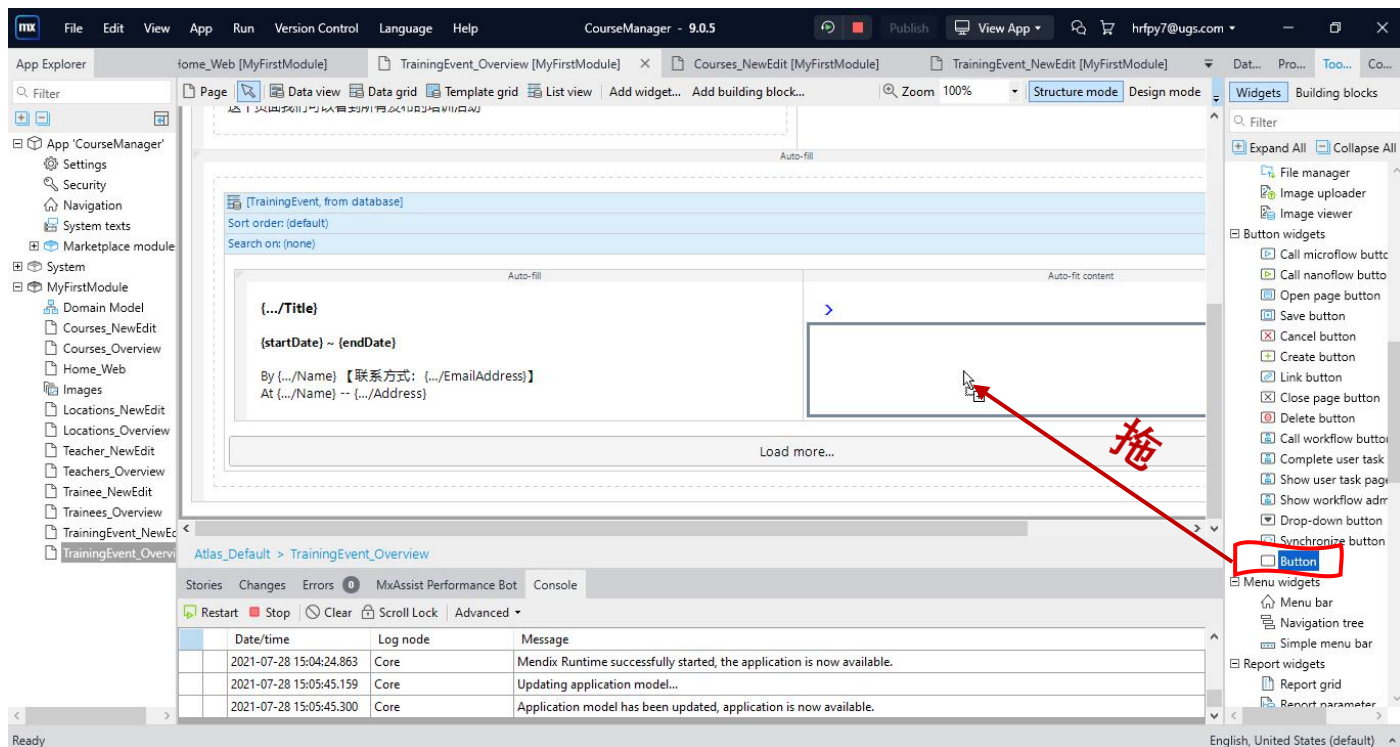
更多请参见: [microflows#1-introduction](#)

3、微流可以做那些事情？

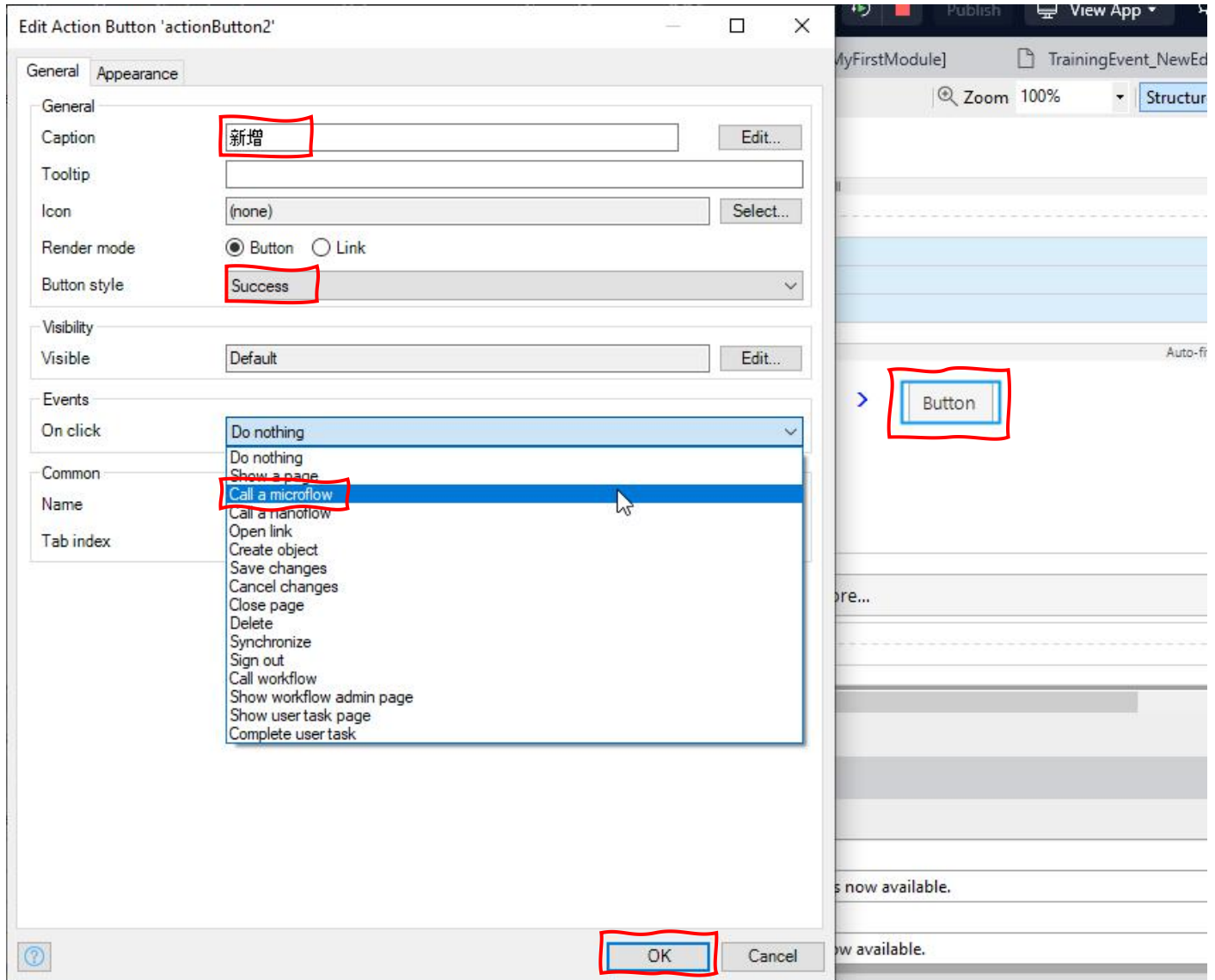
3.1 举例 1（用微流打开新增页面）

基于【Mendix 敏捷开发零基础学习《一》】章节我们继续来学习，我们创建完培训活动的功能模块以后，可以在列表页增加一个新增的按钮。然后调用微流来实现展示页面。接下来开始操作：

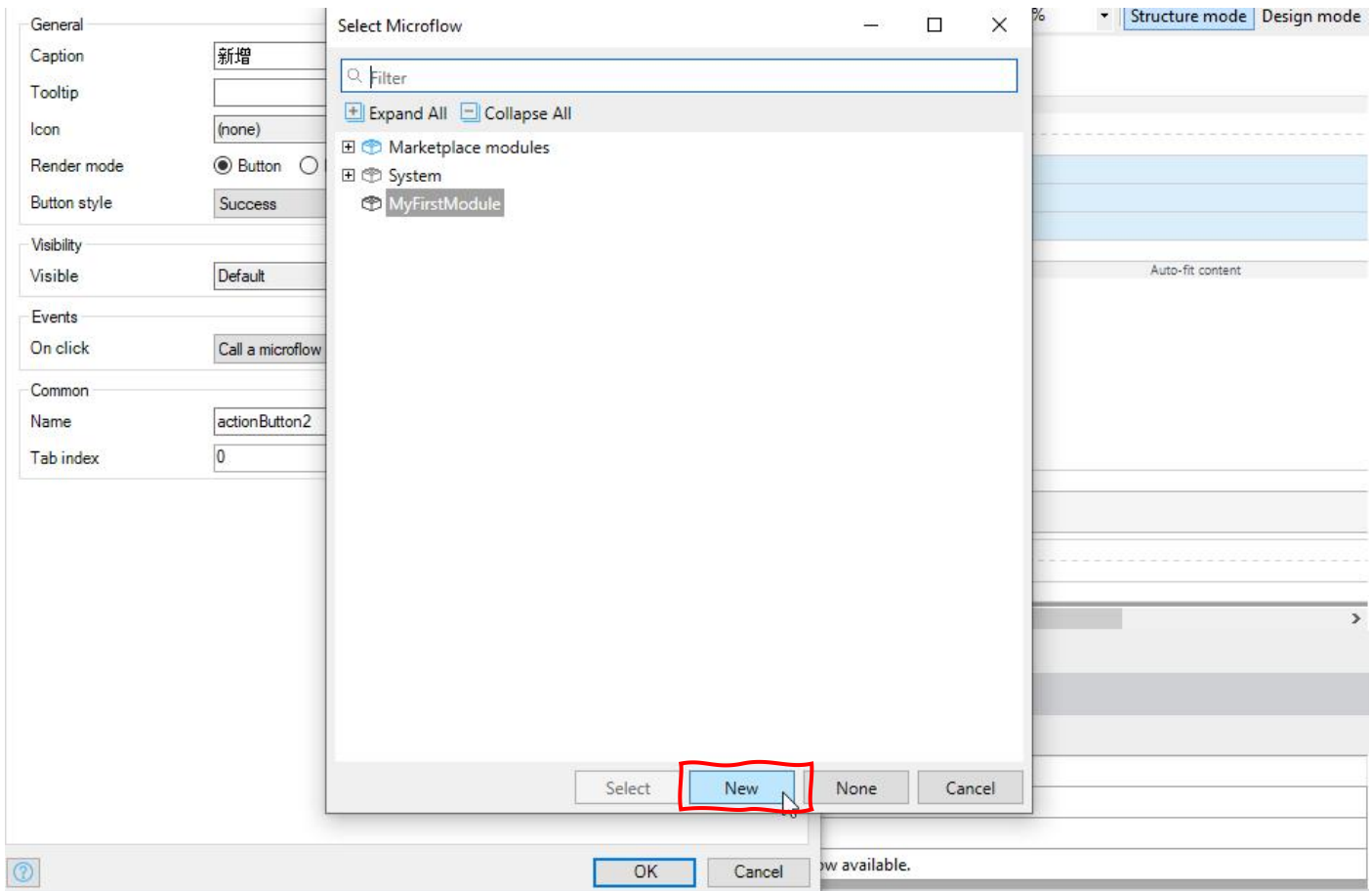
1. 双击 **1 左** 目录树中的 **TrainingEvent_Overview** 培训活动列表页，在 **3 右** 的 **Toolbox** 中的 **widgets** 搜索框中搜索 **button**，将空白 **button** 拖入 到右边列。



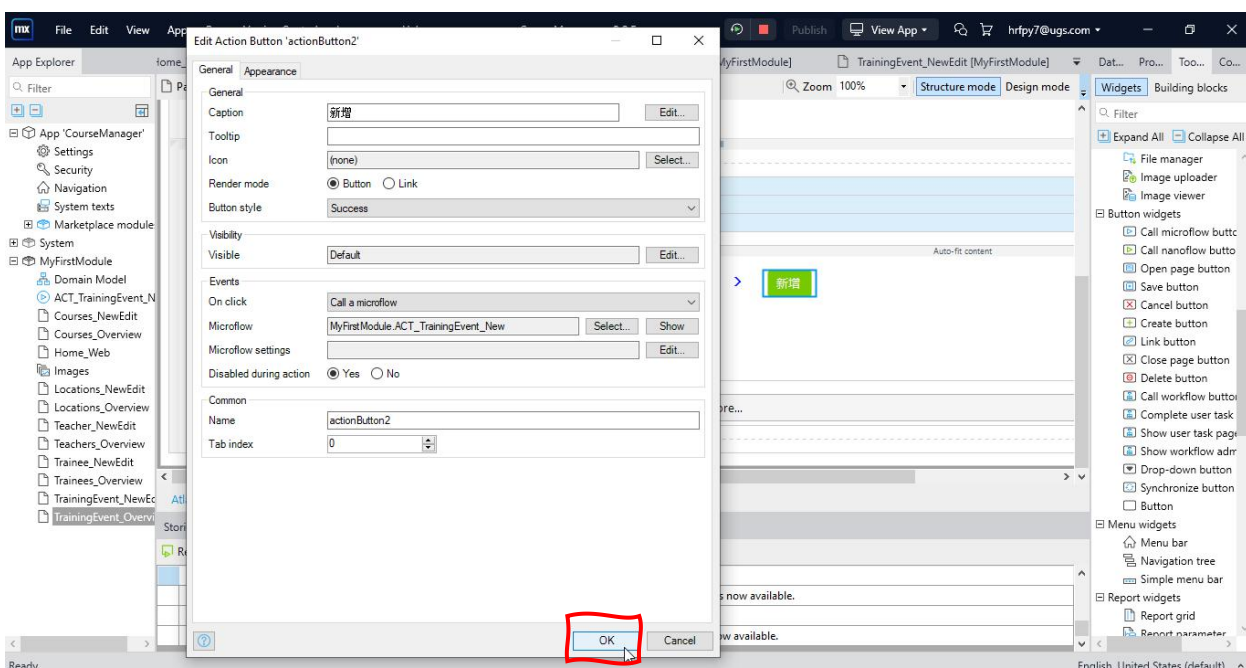
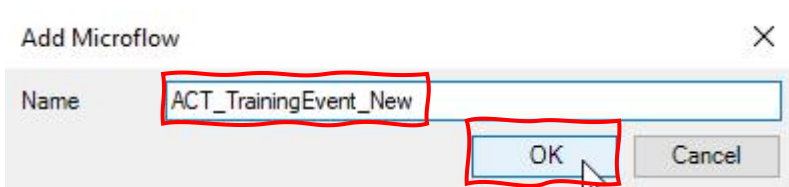
2. 双击拖入的按钮 Button，修改 **Caption**(按钮名称) 为**新增**，修改 **On click** 下拉框为 **Call a microflow**，创建一个微流。



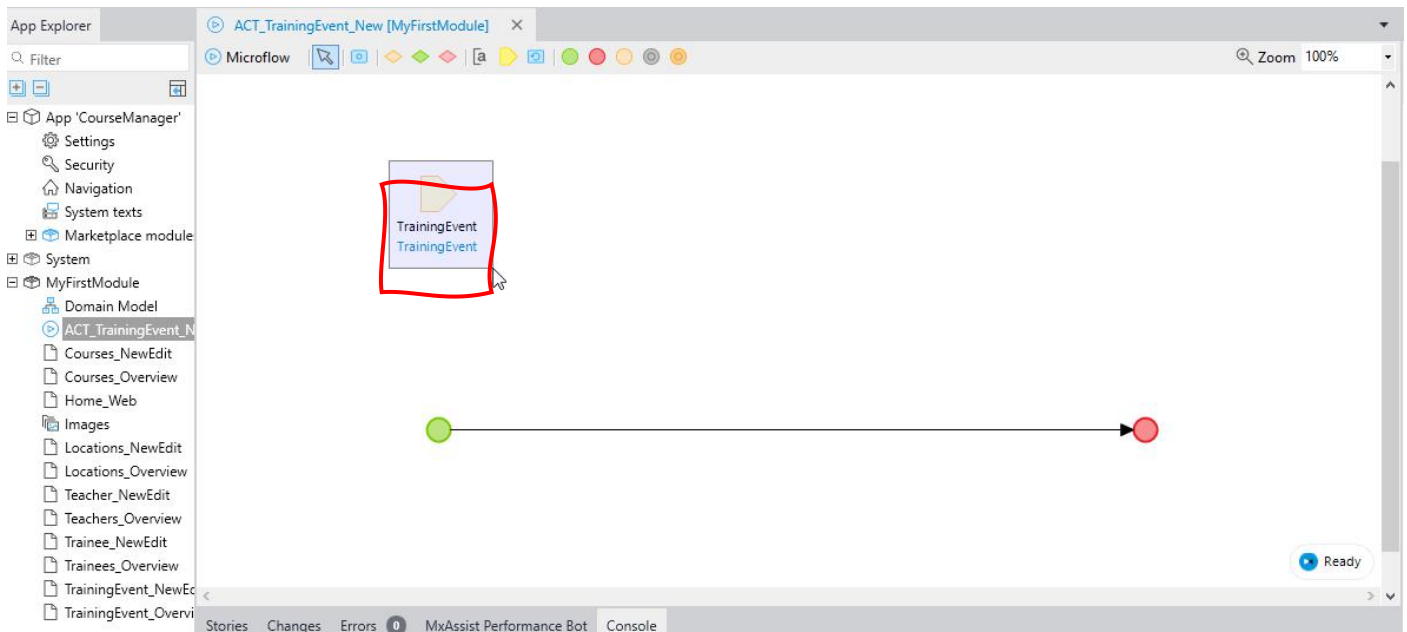
3. 在弹出 Select Microflow 的页面选择 New 按钮，新建微流。



4. 在新弹出的窗口输入 Name 选项 **ACT_TrainingEvent_New**（遵循微流命名规范），然后点击 **OK->Ok** 按钮，即可完成创建。

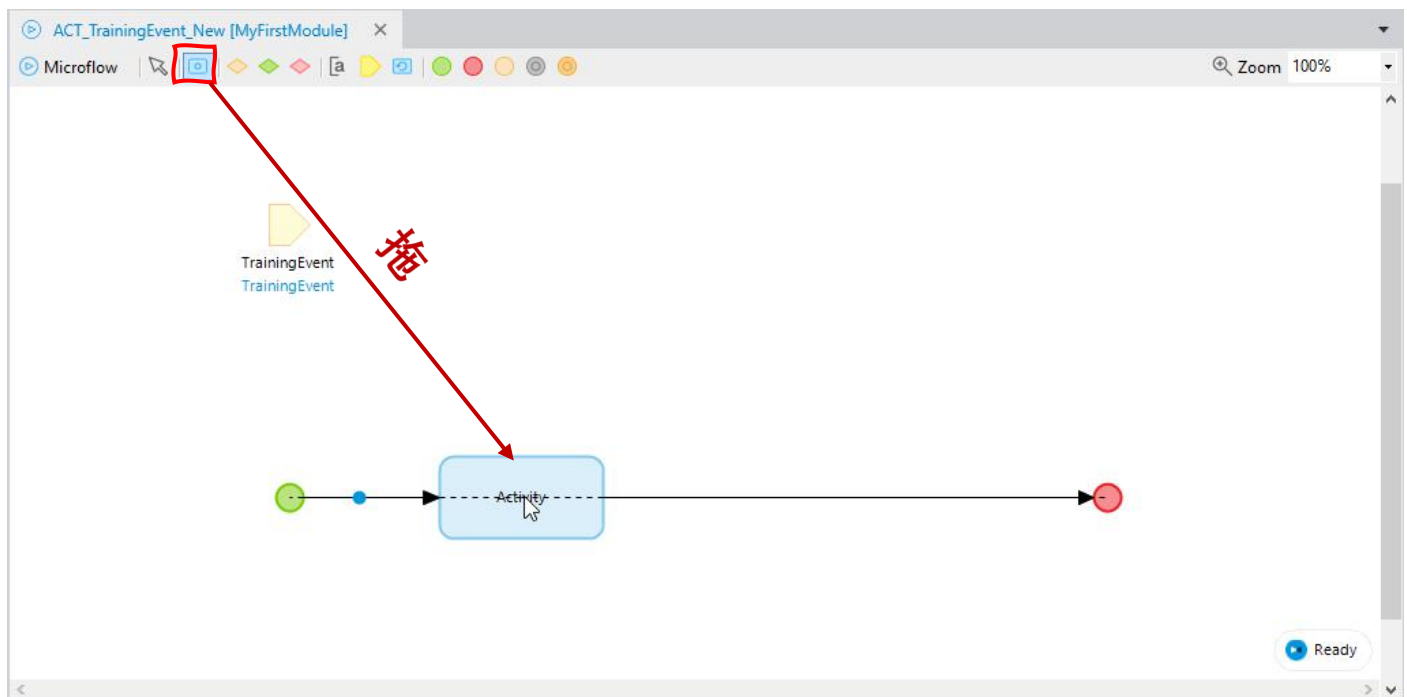


5. 此时目录树中新增了一个微流 **ACT_TrainingEvent_New**。双击该页面，我们可以看到微流的工作台。图中标注内容是刚才创建的新增按钮所在的数据源 **TrainingEvent**，会自动带入到该微流作为常量来使用。

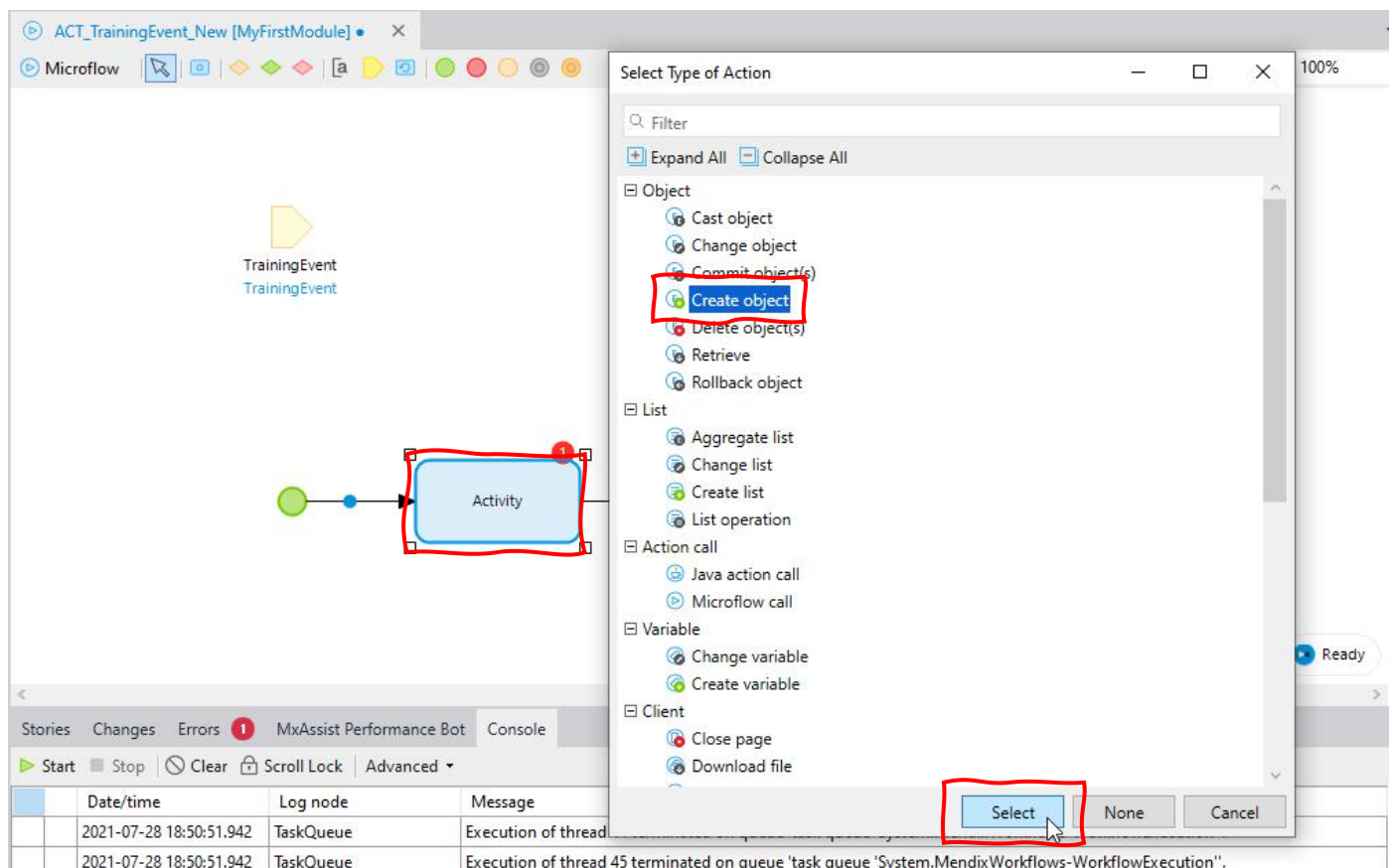


6. 该微流要做的事情：第一，获取 TrainingEvent 数据模型；第二，展示新增页面。

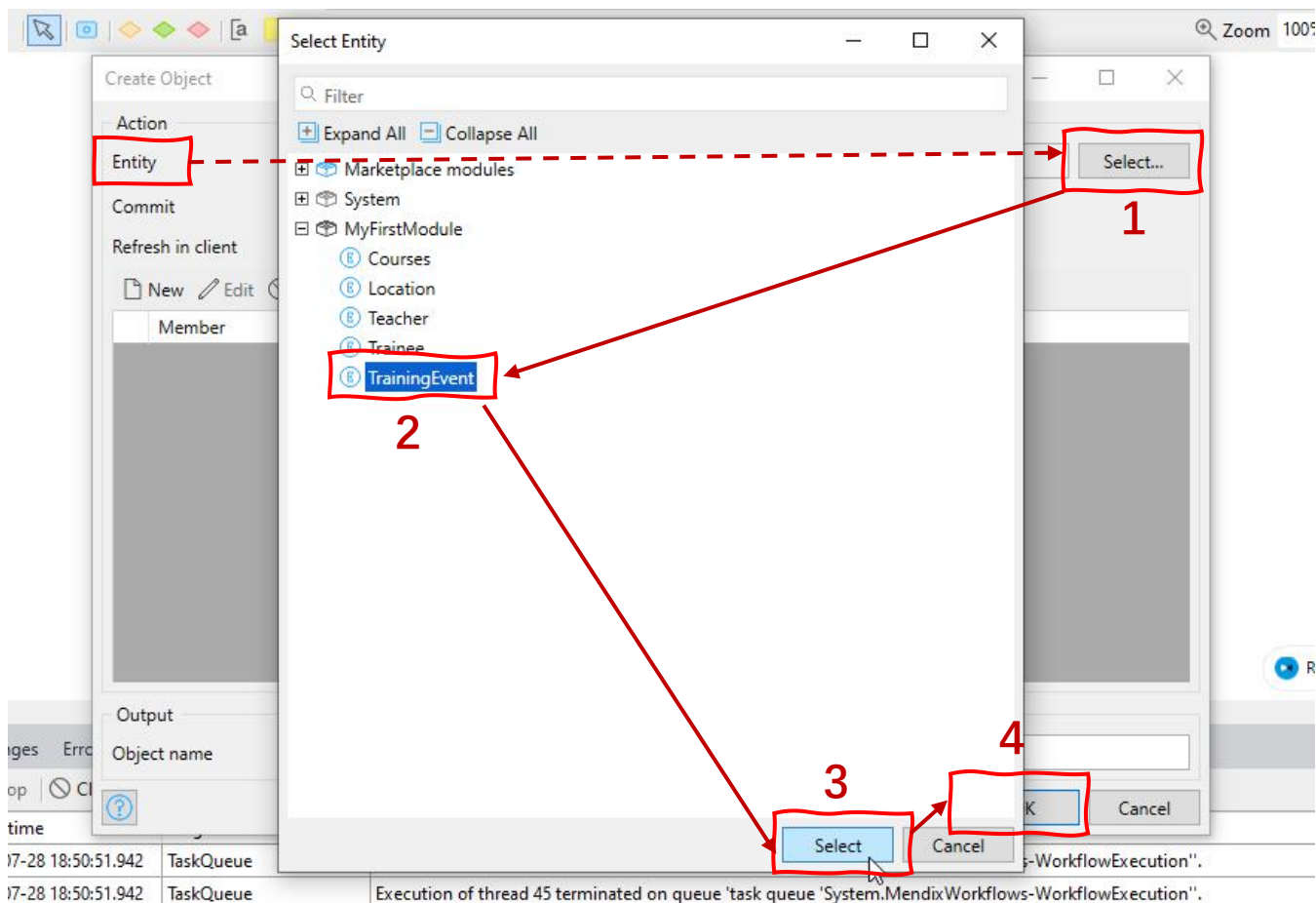
首先鼠标单击 下图圆圈中的任务节点拖入到顺序流中。



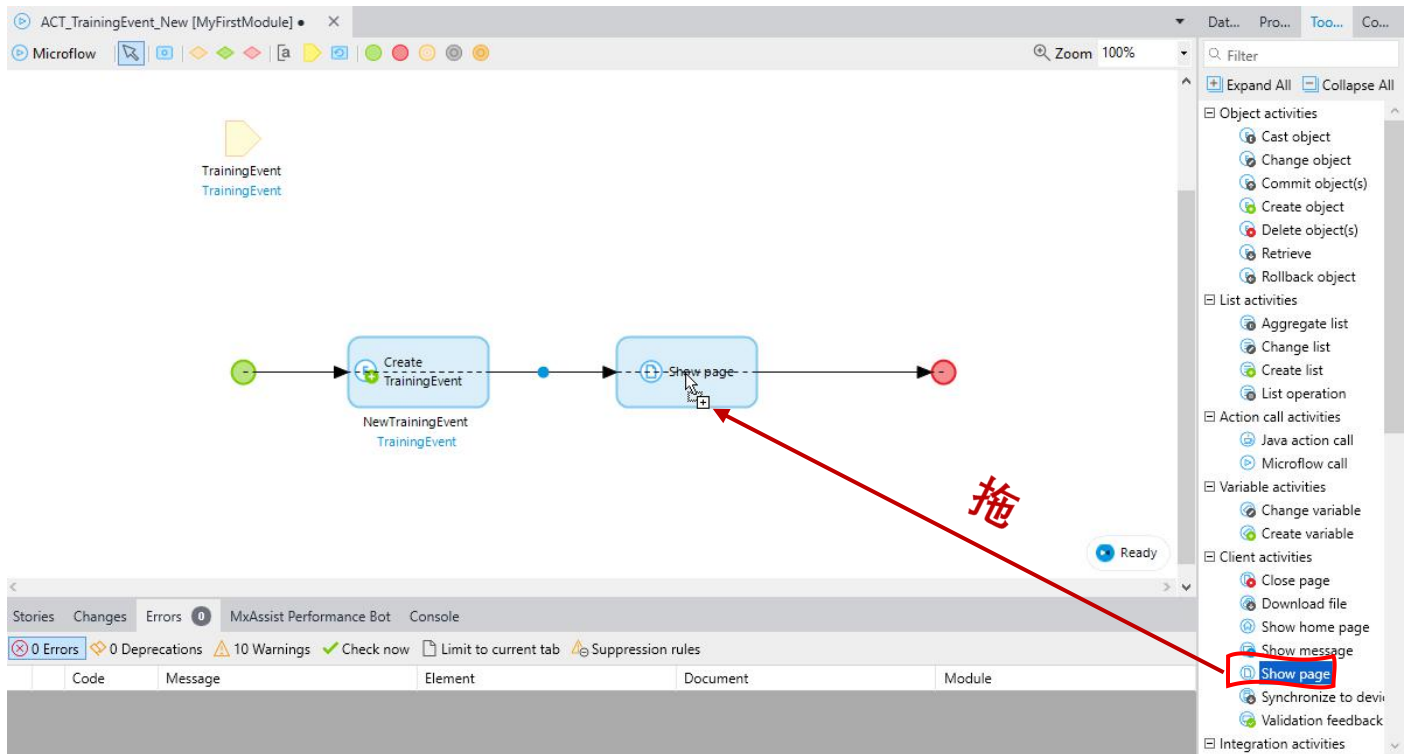
7. 双击刚才拖入的任务节点，选择 Object 下的 Create object，点击 Select 按钮。



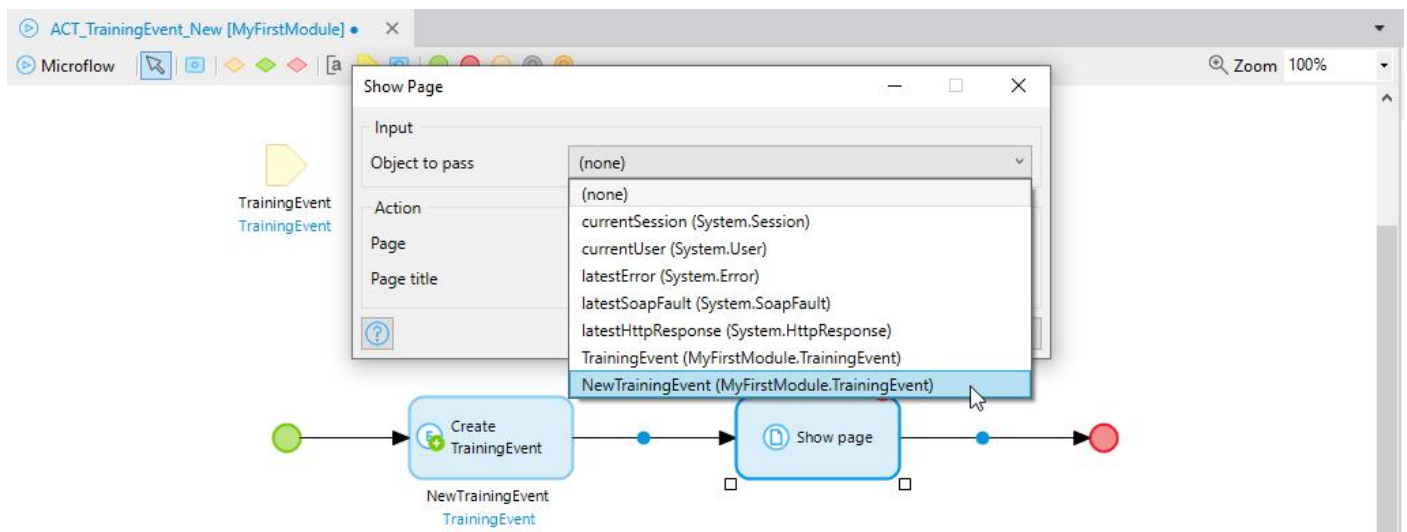
8. 在弹出的 Create Object 窗口中点击 Entity 栏的 Select 按钮，选择 TrainingEvent 数据源，点击 Select->Ok 按钮，完成这个动作 activity。



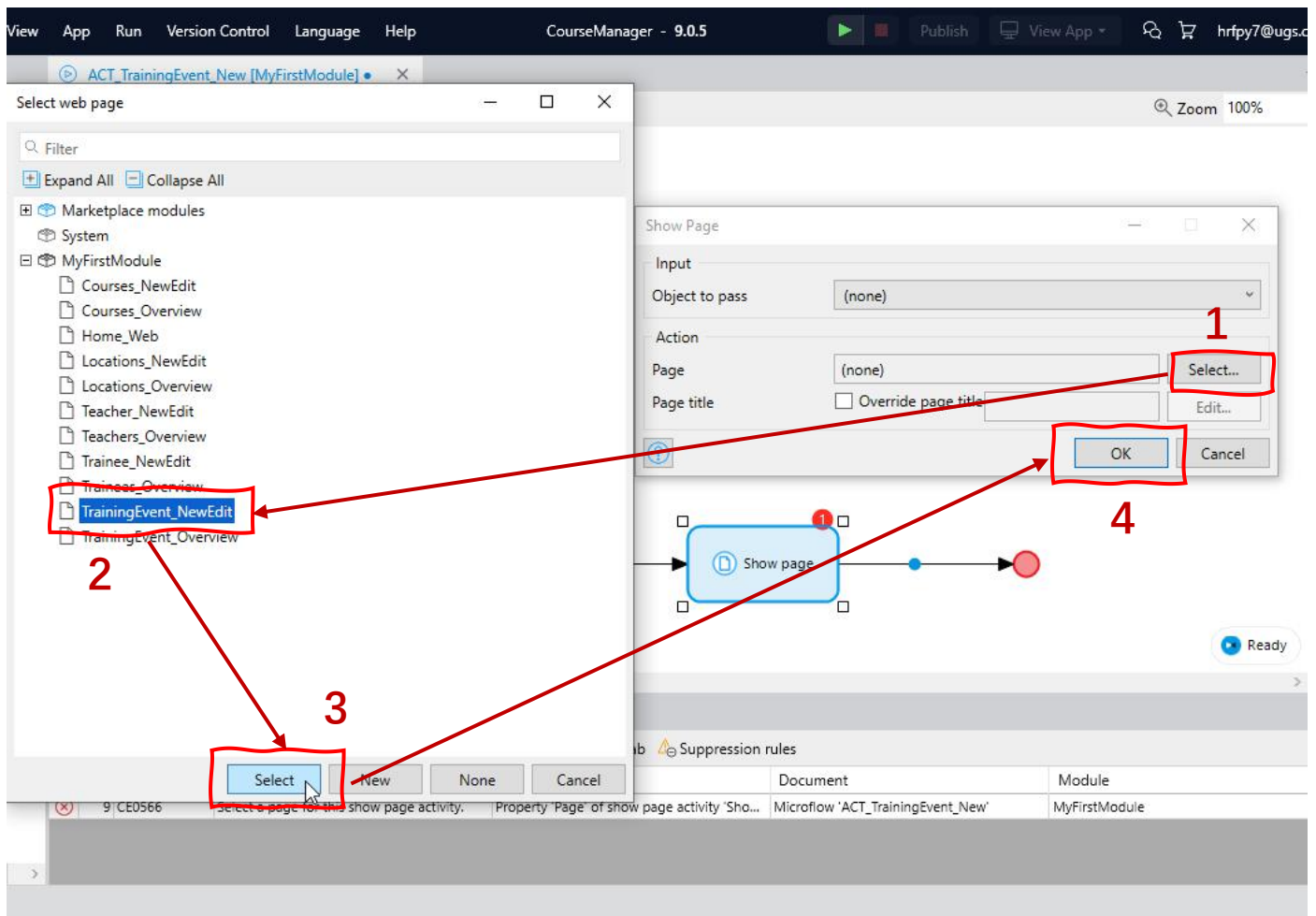
9. 接下来，需要一个新增页面(创建培训活动)：从 3 右把 Client activities 下的 Show page 动作拖入微流中。



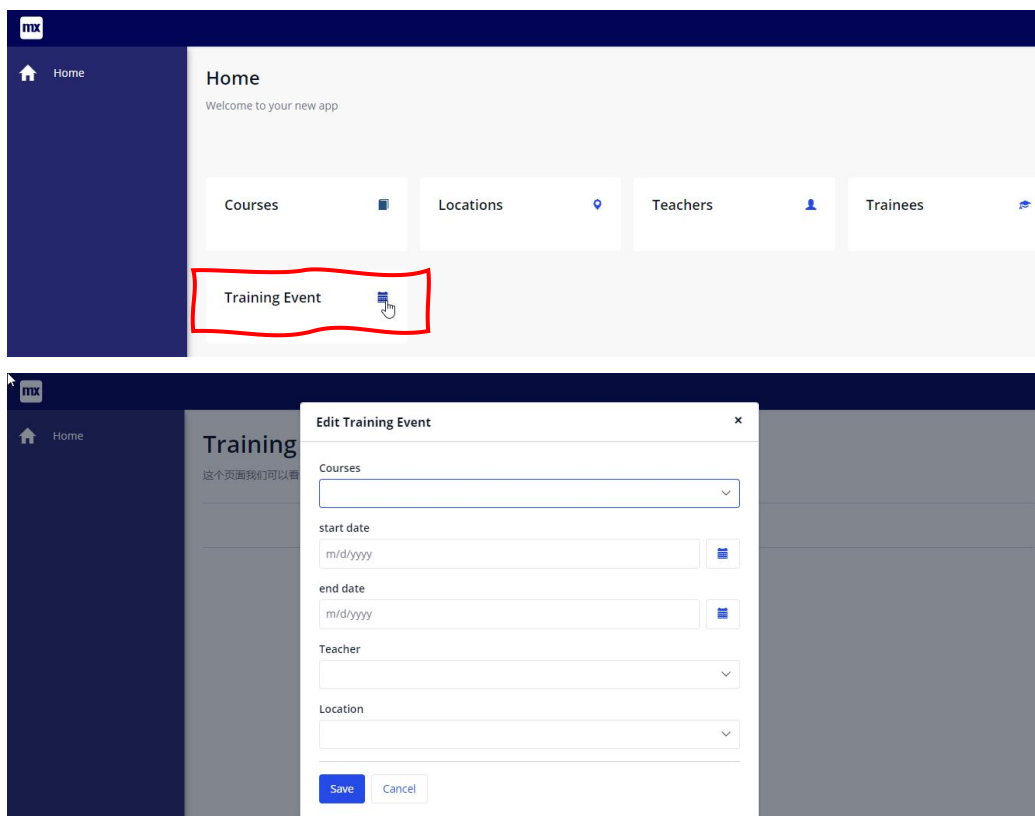
10. 双击 show page 活动块，弹出的窗口中 Object to pass 下拉框中选择前个模块获取的数据节点 NewTrainingEvent。



11. Page 栏点击 **Select** 选择需要显示的页面：**新增页面--TrainingEvent_NewEdit**，点击 **Select->Ok** 按钮。



12. 运行项目，点击 **View** 可以预览如何通过微流打开创建的页面。点击菜单栏中的 **TrainingEvent**，点击新增按钮，这时候我们发现，已经可以用微流打开页面了。



3.2 举例 2（用微流做运算、触发器）

在开始报名数量统计之前，请先完成【三、数据嵌套】

通过**举例 1** 我们接触到微流的应用场景，其实微流还可以做很多事情。我将一些常用的函数和运算公式罗列一下：

函数	备注	输入	输出
*	乘法运算	2*3	6
Div	除法运算	3 div 5	0.6
:	除法运算	12 : 3	4.0
Mod	计算一个数除以另一个的余数	23 mod 5	3
+	加法运算	-3 + 4	1
-	减法运算	5 - 4	1
parseInteger	接受字符串并将其解析为整数。	parseInteger('42')	42
toString	将指定的值转换为字符串表示形式。	toString(1.4)	'1.4'
addDays	在日期上增加几天。	addDays(dateTime(2007, 1, 1, 1, 1, 1), 3)	"Mon Jan 04 01:01:01 CET 2007"
addWeeks	在日期上增加几周。	addWeeks(dateTime(2007, 1, 1, 1, 1, 1), 2)	"Mon Jan 15 01:01:01 CET 2007"
addMilliseconds	为日期添加指定的毫秒数。	addMilliseconds(dateTime(2007, 1, 1, 1, 1, 1), 1400)	"Mon Jan 01 01:01:02:400 CET 2007"
addSeconds	为日期添加指定的秒数。	addSeconds(dateTime(2007, 1, 1, 1, 1, 1), 2)	"Mon Jan 01 01:01:03 CET 2007"
addMinutes	为日期增加了一些分钟。	addMinutes(dateTime(2007, 1, 1, 1, 1, 1), 3)	"Mon Jan 01 01:04:01 CET 2007"
addHours	为日期增加了几个小时。	addHours(dateTime(2007, 1, 1, 1, 1, 1), 25)	"Mon Jan 02 02:01:01 CET 2007"

更多请参考官方：[Expressions](#)

参考完函数和运算公式表，请【跳回 [addDays](#) 讲解】

在开始报名数量统计之前，请先完成【三、数据嵌套】

3.2.1 报名数量统计

3.2.1.1 业务需求

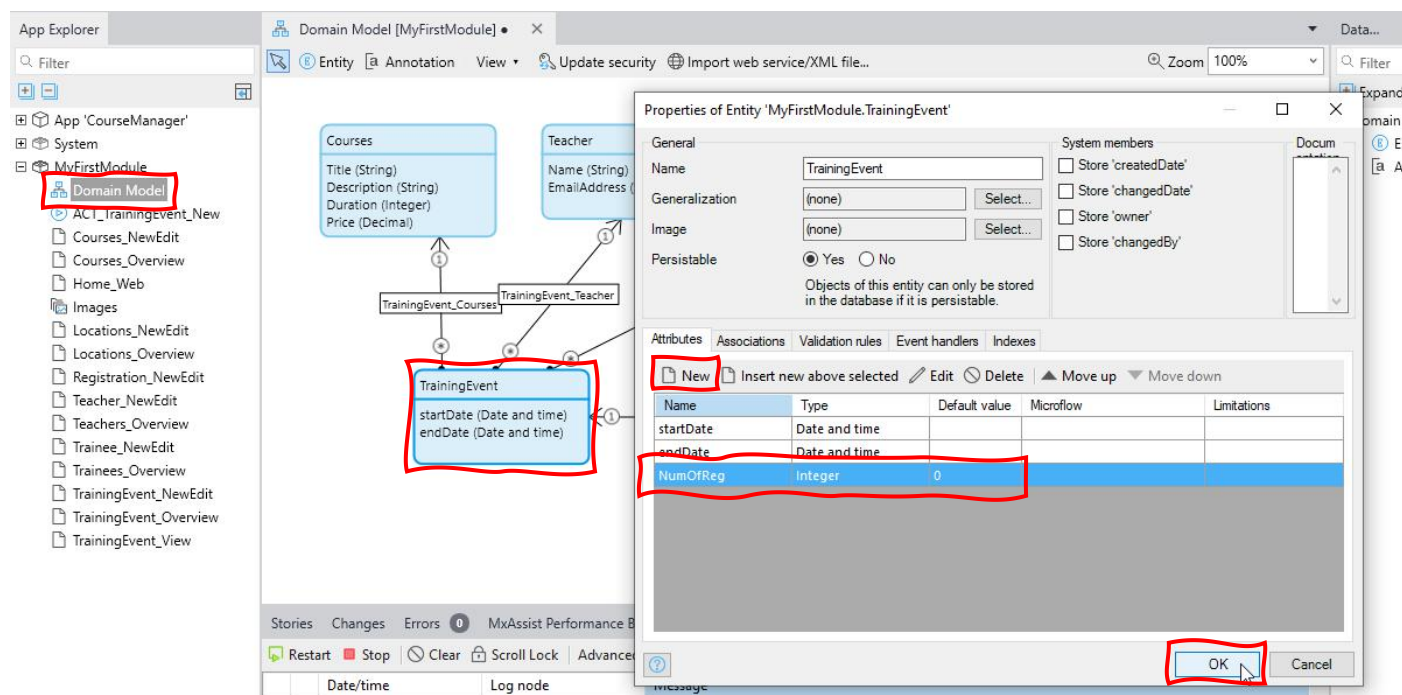
现在有一个新需求，**管理员需要在活动信息报名模块，展示每一个活动报名了多少人。**

3.2.1.2. 需求分析

根据分析可以分为几部分：第一部分，用户报名时，数值增加 1；第二部分，用户取消报名时，数值减少 1；第三部分，将数值展示到页面。

3.2.1.3. 项目实施

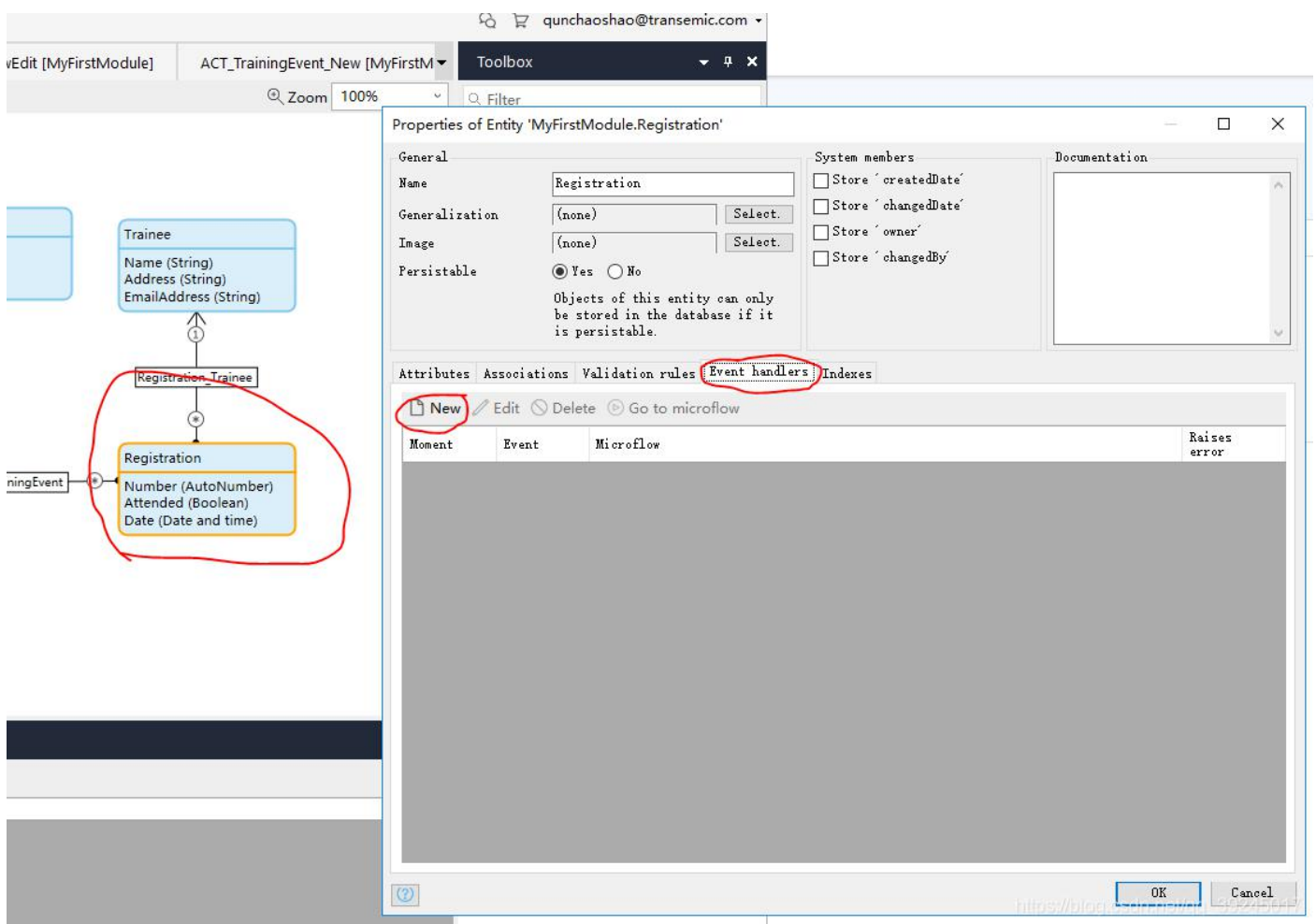
1. 在**活动详情(TrainingEvent)**的数据模型中追加字段 **NrOfRegistration**，记录报名人数，数据类型 **Type** 为 **Integer**，**Default value** 默认值为 **0**。



2. 然后我们需要在报名人员新增和删除的时候去更新 TrainingEvent 数据模型中的 NrOfRegistrations 数量字段。

我们双击报名人员 Registration 数据模型，然后找到 Event Handlers，点击 New 按钮。

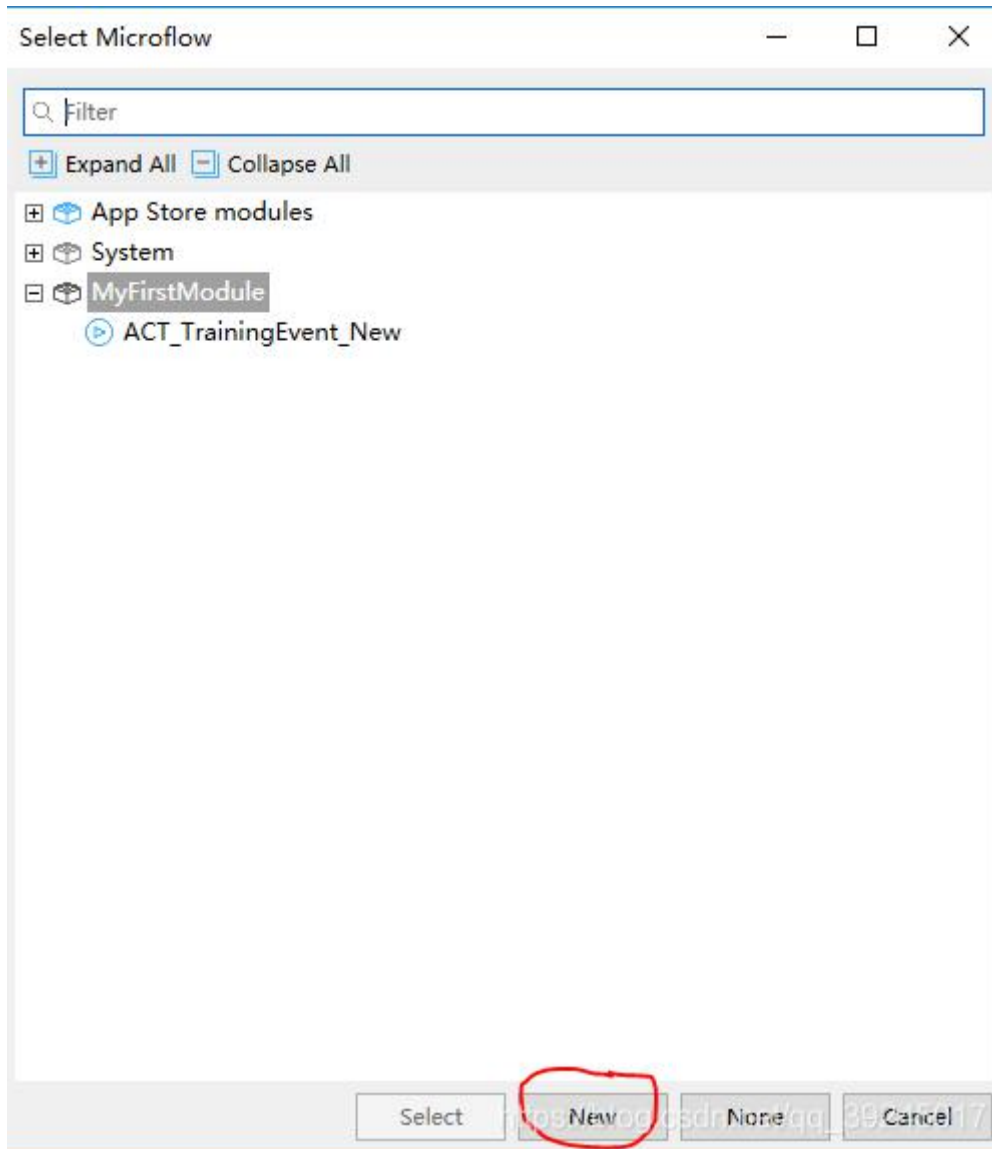
在开始报名数量统计之前，请先完成【三、数据嵌套】，其中有关 于 Registration 构建的详细教程



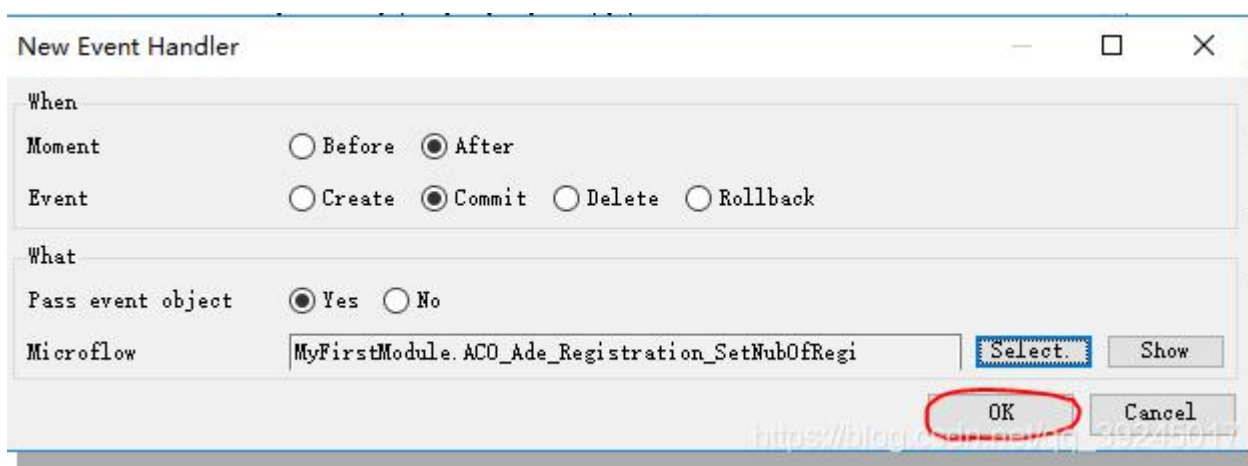
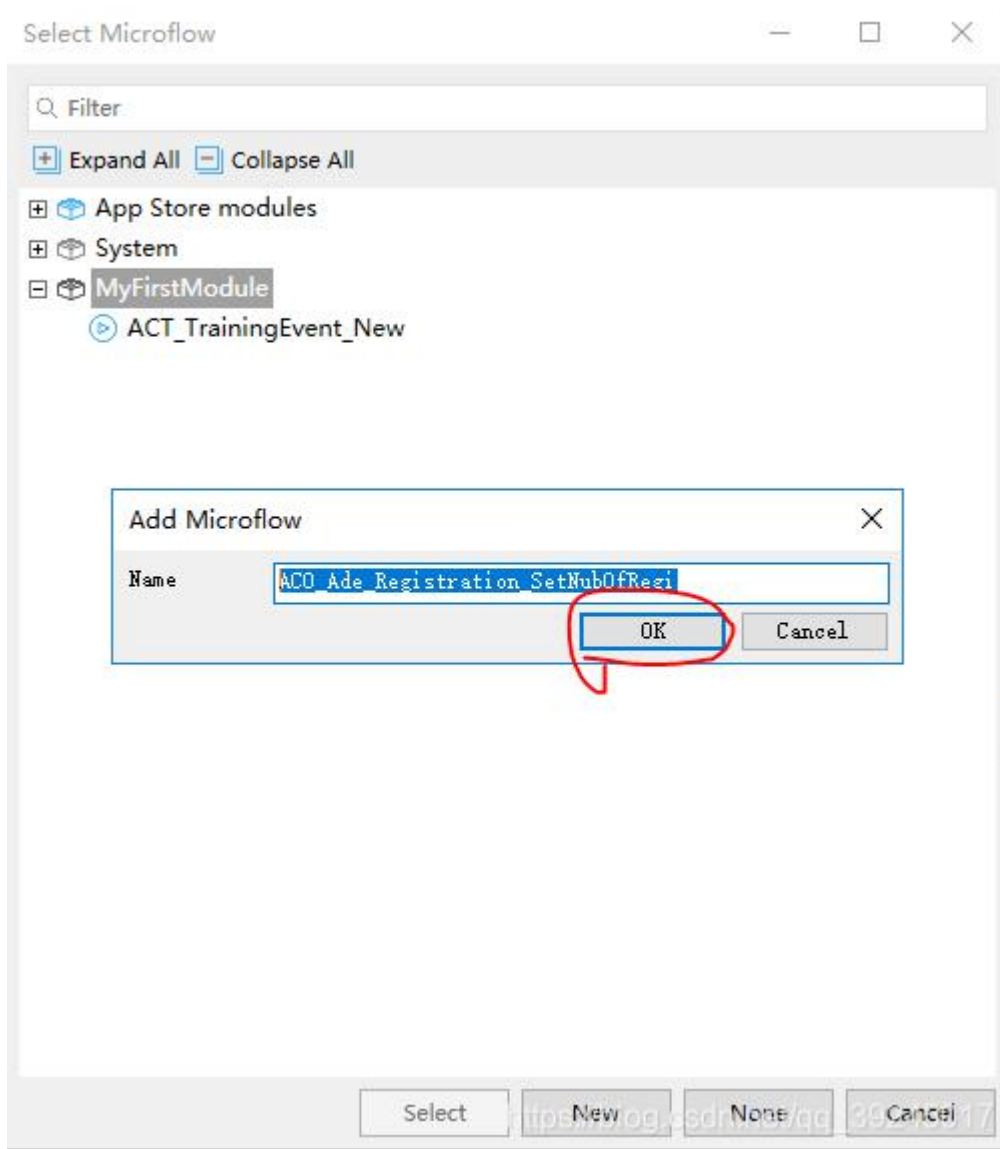
3. 因为我们要求的逻辑时表单提交以后更新，所以 Moment 选择 After(**After** 是在什么操作之后，**Before** 是在什么操作之前)，Event 事件我们选择 Commit(**Create** 是创建事件、**Commit** 是表单提交事件、**Delete** 是删除事件、**Rollback** 是数据回滚事件)，然后选择 Select 按钮选择 Microflow 微流。



4. 因为我们这个时候还没有 Microflow 微流，所以需要 New 新创建一个。



5. 在弹出的页面输入微流的名称 ACO_Ade_Registration_SetNubOfRegi(命名规范), 然后点击 Ok->Ok 按钮。



6. 然后点击 New, 再创建一个数据删除更新的触发器。我们 Moment 选择 After, Event 事件选择 Delect。然后点击 Select 按钮选择我们刚才创建的微流 ACO_Ade_Registration_SetNubOfRegi 然后选择 Select 按钮完成选择。

Properties of Entity 'MyFirstModule.Registration'

General		System members	Documentation
Name	Registration	☐ Store 'createdDate' ☐ Store 'changedDate' ☐ Store 'owner' ☐ Store 'changedBy'	
Generalization	(none) Select.		
Image	(none) Select.		
Persistable	☒ Yes ☐ No Objects of this entity can only be stored in the database if it is persistable.		

Attributes Associations Validation rules Event handlers Indexes

Moment	Event	Microflow	Raises error
After	Commit	MyFirstModule.ACO_Ade_Registration_SetNubOfRegi	

New Event Handler

When

Moment ☐ Before ☒ After

Event ☐ Create ☐ Commit ☒ Delete ☐ Rollback

What

Pass event object ☒ Yes ☐ No

Microflow (none) Select.

OK Cancel

<https://blog.csdn.net/q139241017>

Select Microflow

Filter

+ Expand All - Collapse All

+ App Store modules

+ System

+ MyFirstModule

- ▶ ACO_Ade_Registration_SetNubOfRegi
- ▶ ~~ACT_TrainingEvent_New~~

Select New None Cancel

<https://blog.csdn.net/q139241017>

7. 然后点击 Ok-Ok 按钮完成创建。

New Event Handler

When

Moment ☐ Before ☒ After

Event ☐ Create ☐ Commit ☒ Delete ☐ Rollback

What

Pass event object ☒ Yes ☐ No

Microflow

Properties of Entity 'MyFirstModule.Registration'

General

Name

Generalization

Image

Persistable ☒ Yes ☐ No

Objects of this entity can only be stored in the database if it is persistable.

System members

☐ Store 'createdDate'

☐ Store 'changedDate'

☐ Store 'owner'

☐ Store 'changedBy'

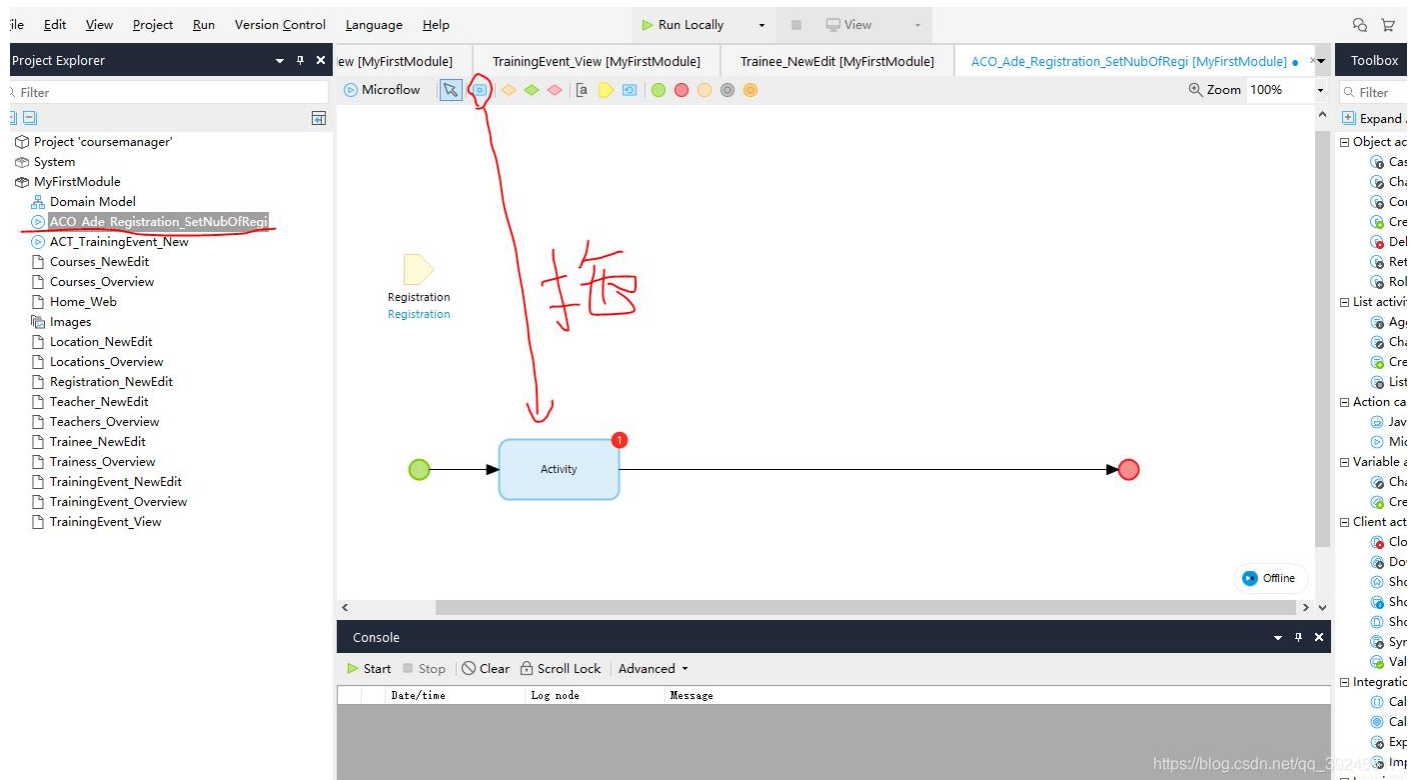
Documentation

Attributes Associations Validation rules **Event handlers** Indexes

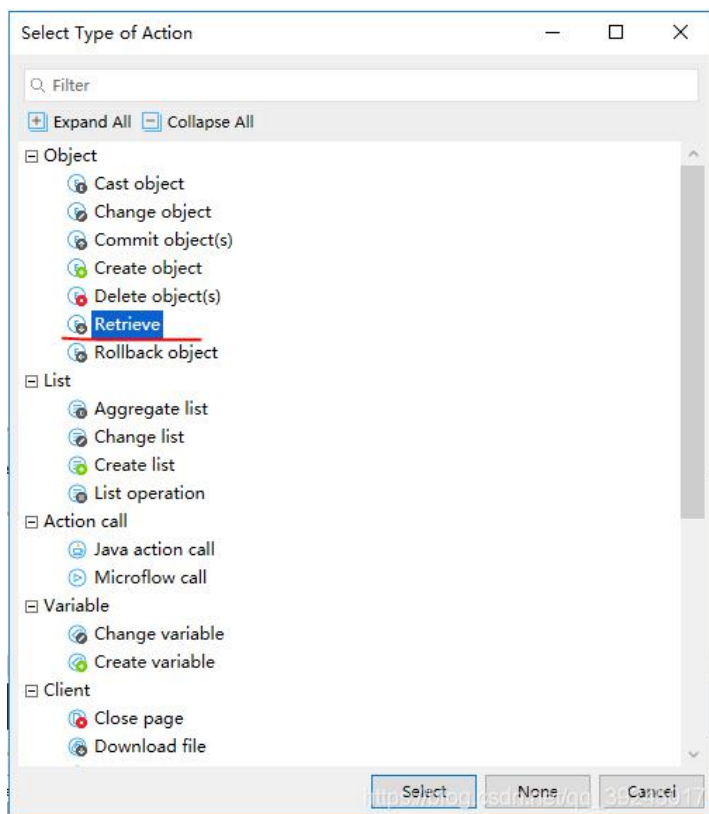
Moment	Event	Microflow	Raises error
After	Commit	MyFirstModule.ACO_Ade_Registration_SetNubOfRegi	
After	Delete	MyFirstModule.ACO_Ade_Registration_SetNubOfRegi	

8. 创建好以后我们，我们在目录树上双击打开我们创建的 Microflow 微流

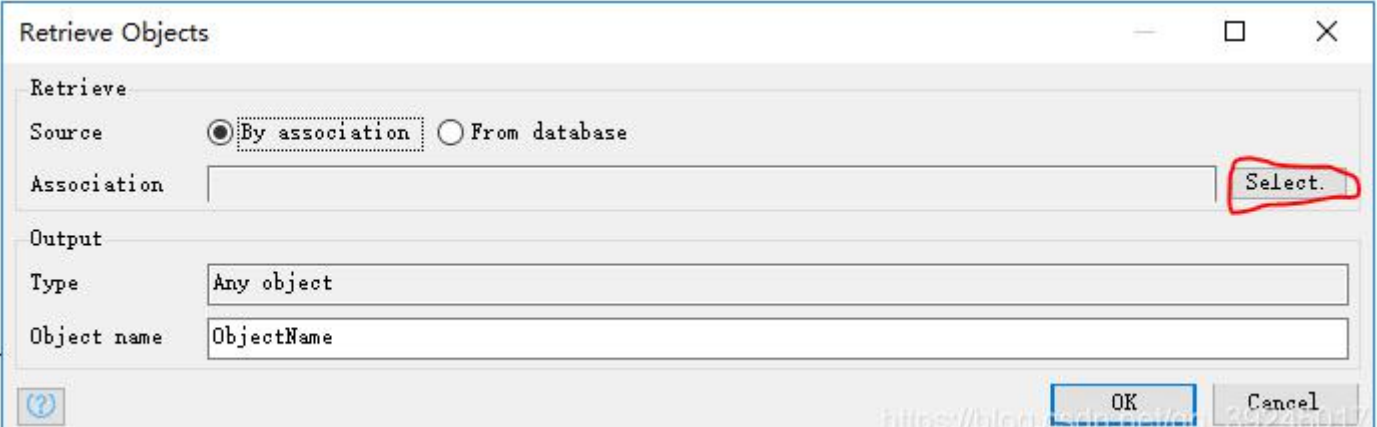
ACO_Ade_Registration_SetNubOfRegi 文件，然后拖入一个 Activity 事件。



9. 然后我们选择 Retrieve(我们最终是需要更改活动 TrainingEvent 数据模型中的 NrOfRegistration 数量字段，所以我们首先需要获取到 TrainingEvent 数据模型)，该类型就是获取缓存或数据库中的数据。然后点击 Select 按钮。

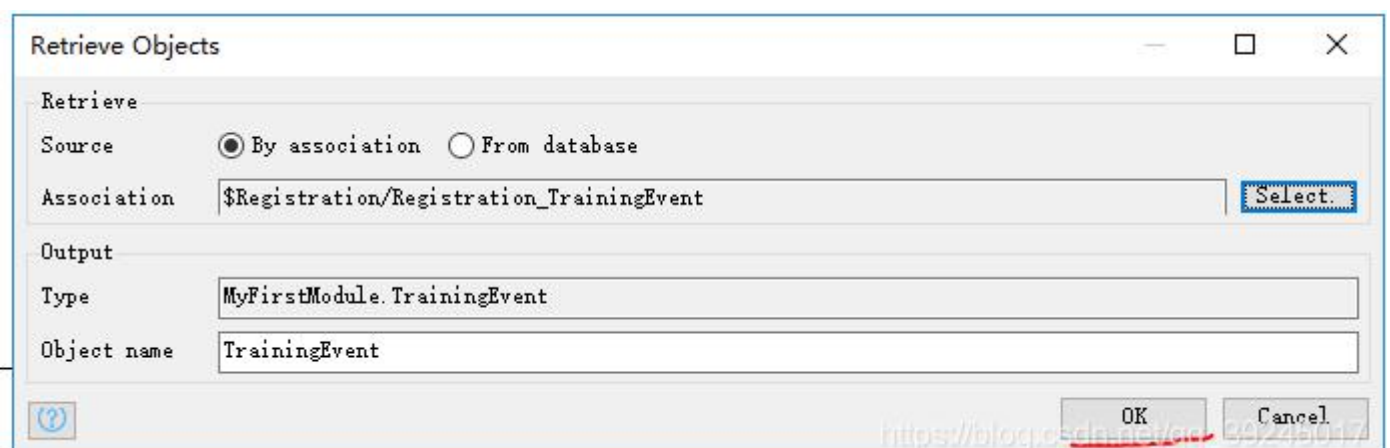
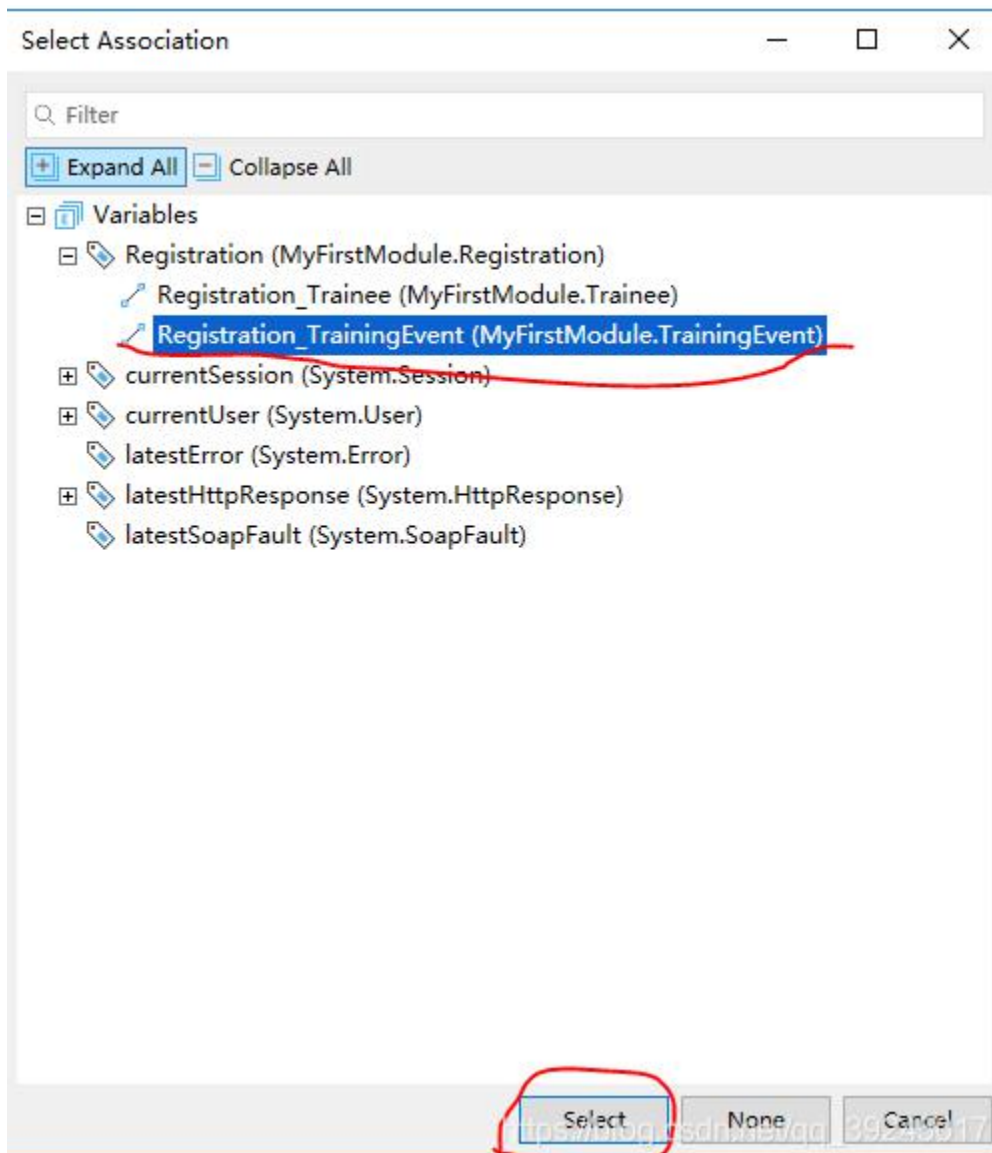


10. 我们在弹出的窗口 Source 选择 By association(By association 是从我们系统内存中获取数据, From database 是从数据库中获取数据), 因为系统运行时会将数据模型中的数据加载到内存中, 所以我们可以直接从内存中获取数据。然后点击 Select 按钮。

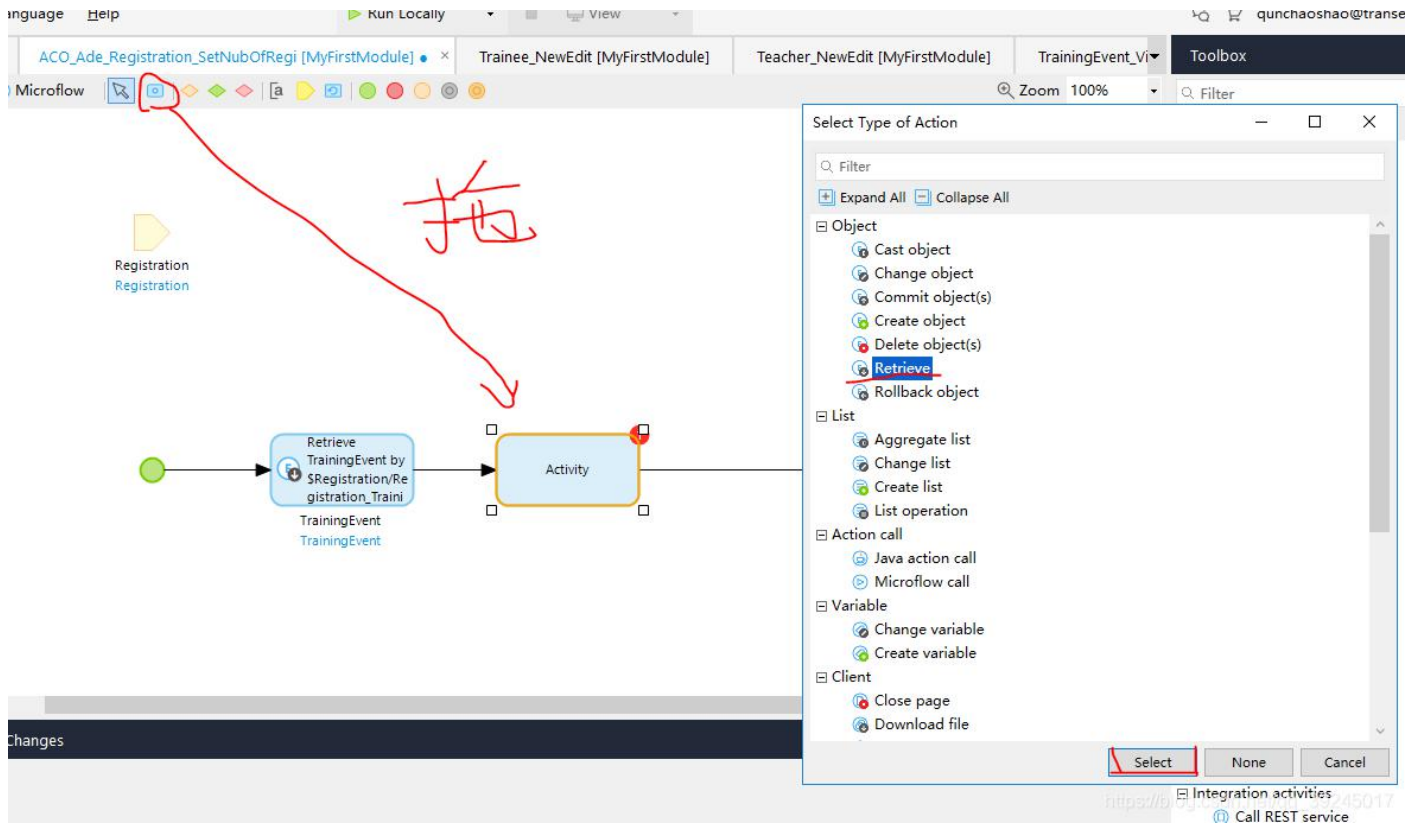


The screenshot shows the 'Retrieve Objects' dialog box. It has a 'Retrieve' section with 'Source' options: 'By association' (selected) and 'From database'. Below this is an 'Association' text field and a 'Select.' button, which is circled in red. The 'Output' section contains 'Type' (set to 'Any object') and 'Object name' (set to 'ObjectName'). At the bottom right are 'OK' and 'Cancel' buttons. A watermark URL is visible at the bottom right of the dialog.

11. 我们选择于报名人员 Registration 数据模型相关联的报名活动 TrainingEvent 数据模型, 选择 Select 按钮。然后点击 Ok 按钮。



12. 然后我们在拖入一个 Activity 然后选择相同的 Retrieve 事件用来获取我们数据库中的报名人员的所有信息。



13. 我们选择 From database 从数据库中获取数据，然后再选择 Select 按钮。

Retrieve Objects

Retrieve

Source

☐ By association
☒ From database

Entity

MyFirstModule.Registration

Select.

Show

Options

Range

☒ All
☐ First
☐ Custom

XPath constraint

[MyFirstModule.Registration_TrainingEvent = \$TrainingEvent]

Line	Column	Error

Sorting

New

Delete

Move up

Move down

Attribute	Sort order

Output

Type

List of MyFirstModule.Registration

List name

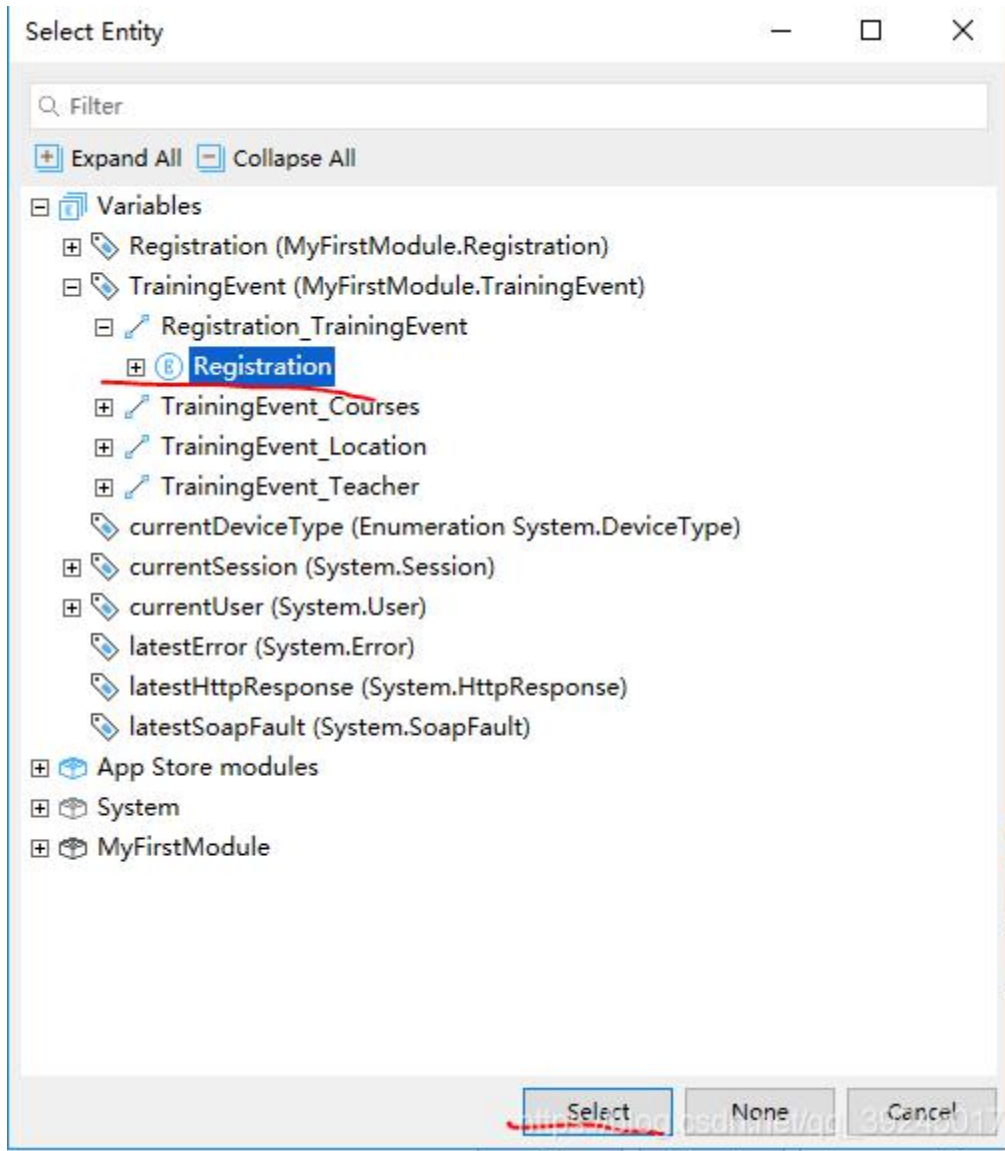
RegistrationList

?

OK

Cancel

14. 然后选择与 TrainingEvent 活动详情相关联的报名人员 Registration 数据模型。然后点击 Select 按钮。



15. 然后点击 Ok 按钮完成创建。

Retrieve Objects

Retrieve
Source ☐ By association ☒ From database

Entity

Options
Range ☒ All ☐ First ☐ Custom

XPath constraint

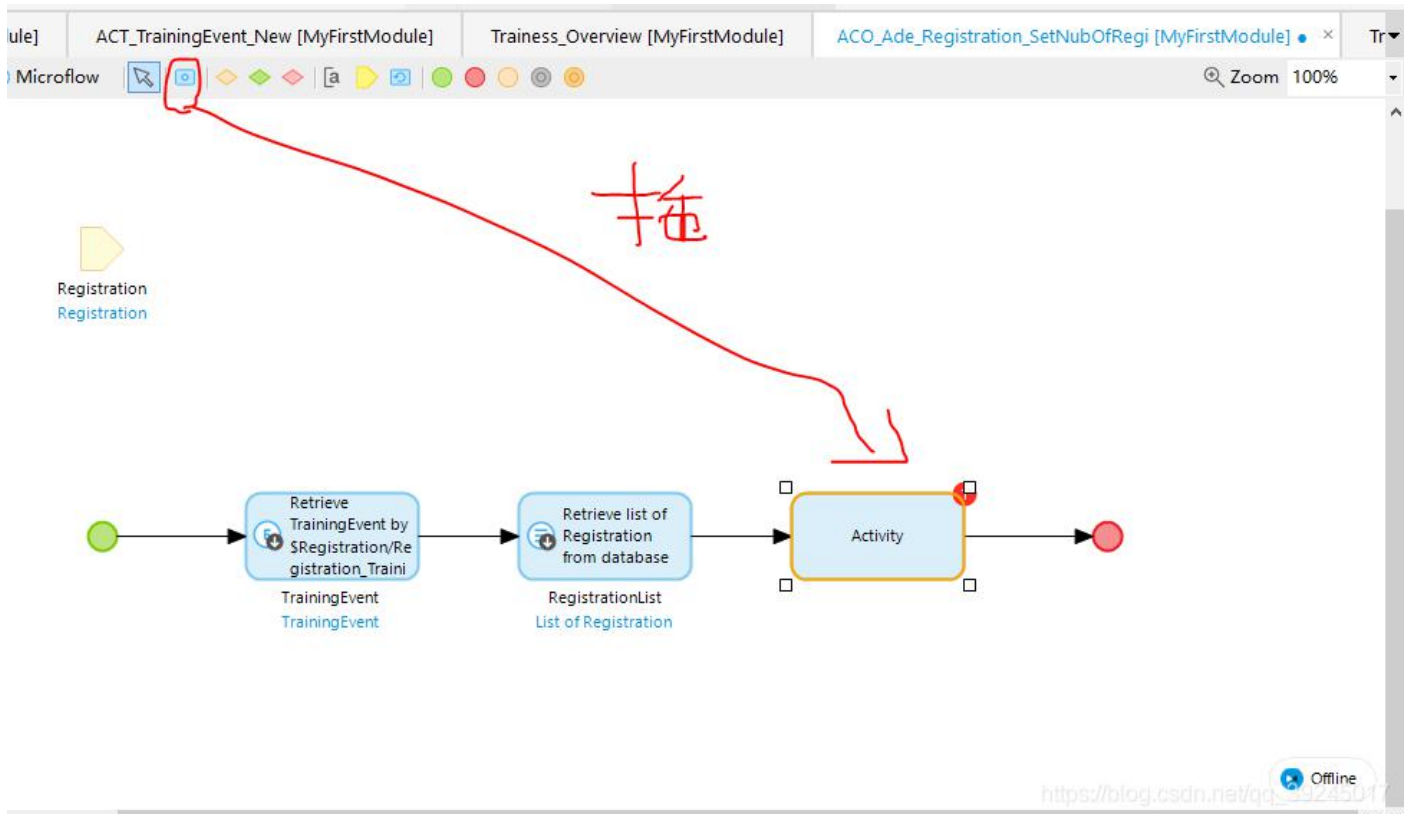
Line	Column	Error

Sorting

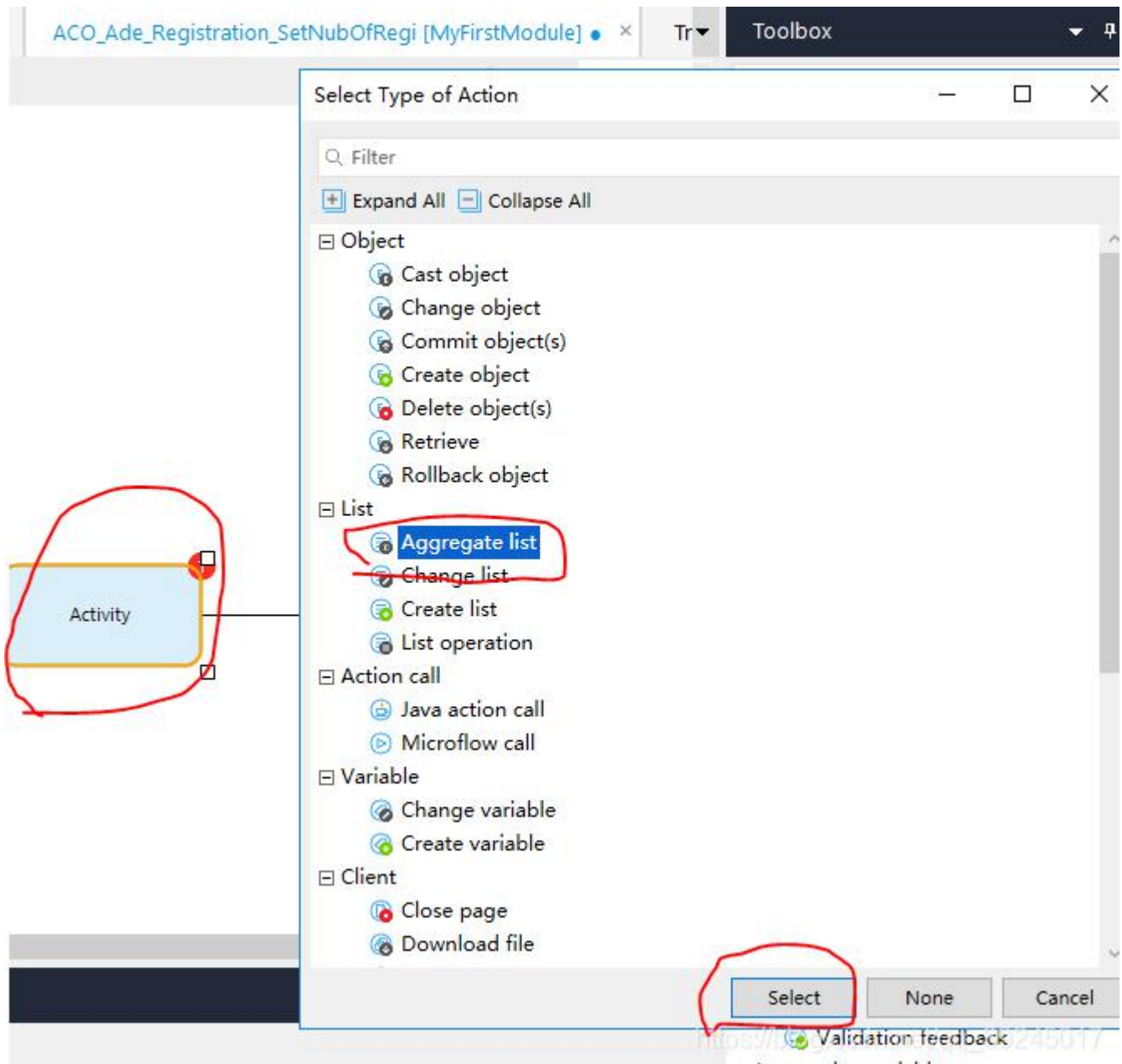
Attribute	Sort order

Output
Type
List name

16. 我们再拖入一个。用来统计刚才获取的所有报名人员。



17. 因为获取到的是一个 list，所以我们双击 Activity 事件选择 Aggregate list，对 List 进行操作。



18. 选择我们刚才获取的报名信息 RegistrationList 数据，然后再 Function 中选择 Count 求和操作。

Aggregate List

Input

List: (none)

Action: RegistrationList (List of MyFirstModule.Registration)

Function: Sum

Attribute:

Output

Variable name: Variable

OK Cancel

Aggregate List

Input

List: RegistrationList (List of MyFirstModule.Registration)

Action:

Function: Sum

Attribute:

Output

Variable name: Variable

OK Cancel

19. 然后我们再 Output 输出的 Variable name 参数 改为 number。然后点击 Ok 按钮。完成操作

Aggregate List

Input

List: RegistrationList (List of MyFirstModule.Registration)

Action:

Function: Count

Attribute:

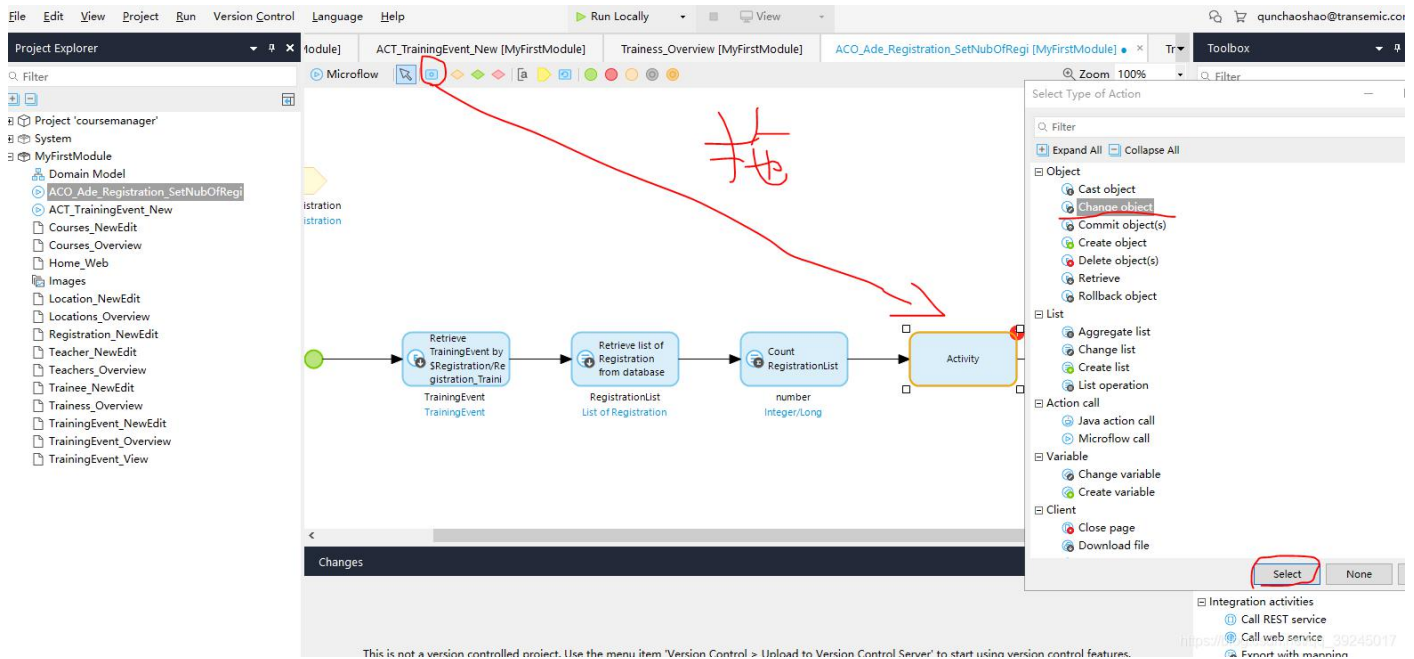
Output

Variable name: number

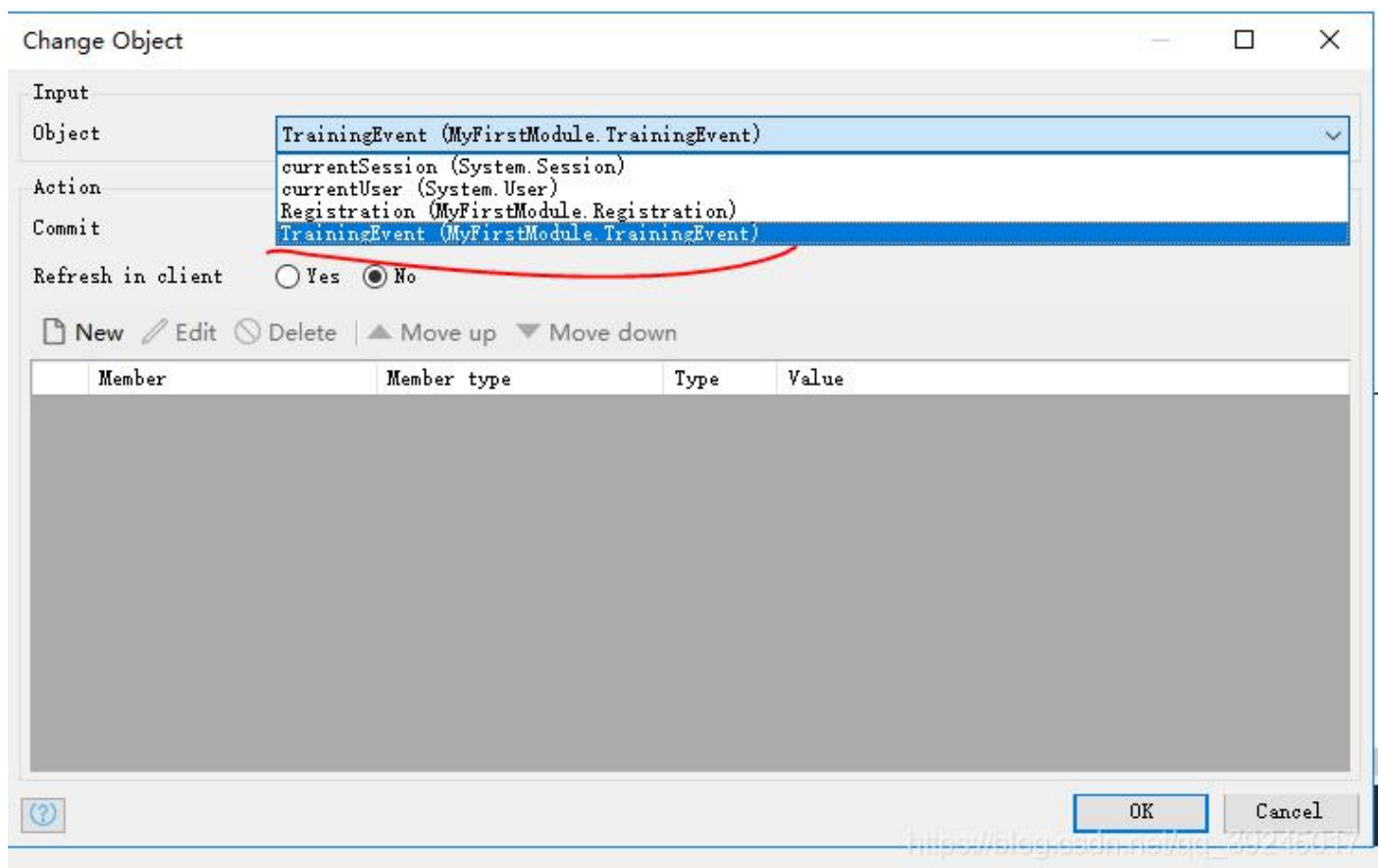
OK Cancel

20. 我们将统计的信息更新到数据库中，所以需要我们在创建一个 Activity 事件并双击，然后在弹出的窗口选择

Change object 更改数据事件，然后点击 Select 按钮。



21. 然后在弹出的页面选择我们第一步获取的 TrainingEvent 活动数据模型。



22. 然后点击 New，编写我们的规则。

Change Object

Input

Object

TrainingEvent (MyFirstModule.TrainingEvent)

Action

Commit

☐ Yes ☐ Yes without events ☒ No

Refresh in client

☐ Yes ☒ No

New

Edit

Delete

Move up

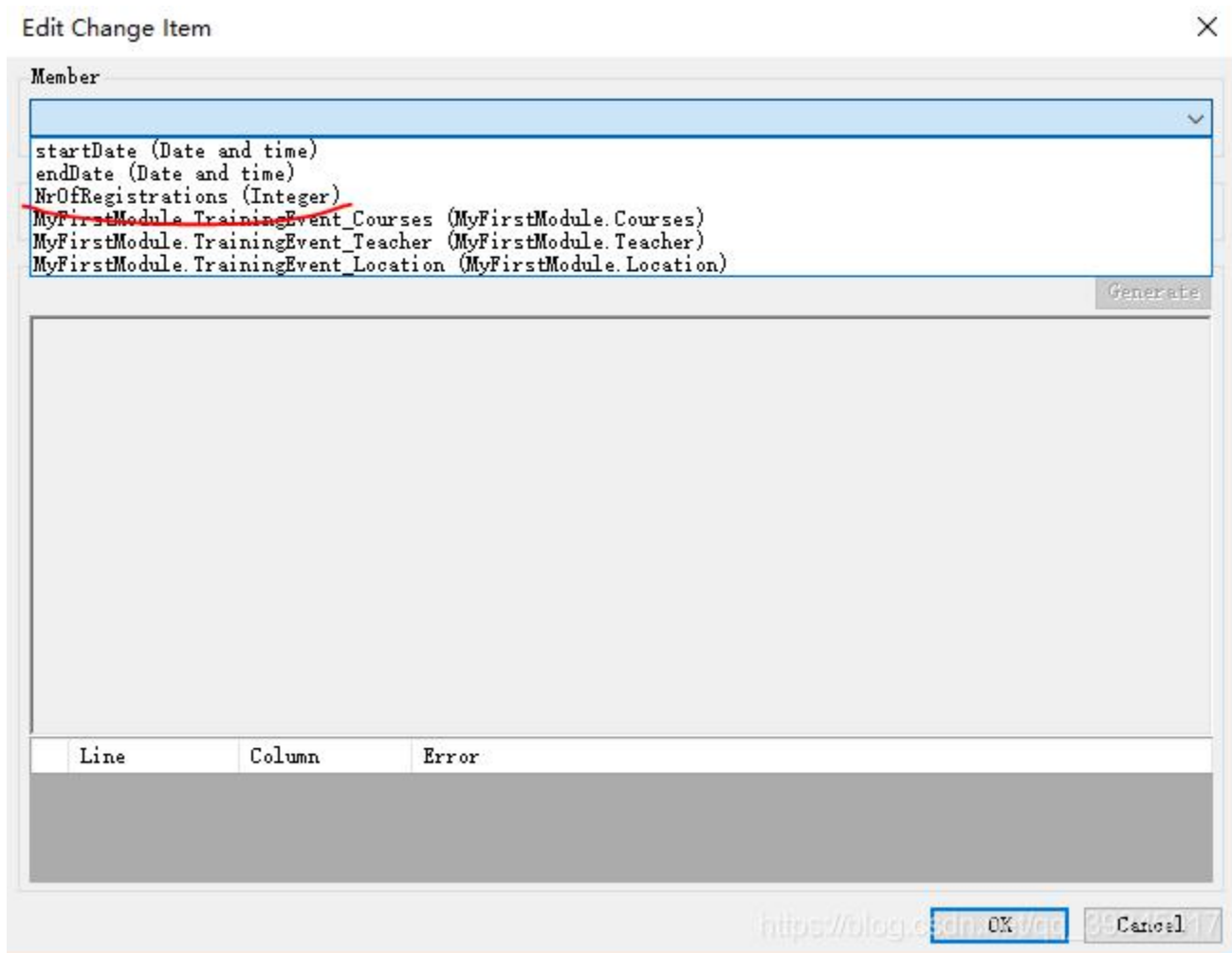
Move down

Member	Member type	Type	Value
--------	-------------	------	-------

OK

Cancel

23. 选择需要修改的字段 NrOfRegistrations。



24. 然后 Type 选择 Set，在输入框中打出\$会自动的带出我们微流的一些信息，其中 number 是我们刚才获取的所有用户的数量。所以选择 number 就可以。选择好以后点击 Ok->Ok 按钮。

Edit Change Item

Member

NrOfRegistrations (Integer) ▼

Type

☒ Set ☐ Add ☐ Remove

Value

Generate

\$

- \$currentDeviceType (Enumeration System.DeviceType)
- \$currentSession (System.Session)
- \$currentUser (System.User)
- \$latestError (System.Error)
- \$latestHttpResponse (System.HttpResponse)
- \$latestSoapFault (System.SoapFault)
- \$number (Integer/Long)
- \$Registration (MyFirstModule.Registration)
- \$RegistrationList (List of MyFirstModule.Registration)
- \$TrainingEvent (MyFirstModule.TrainingEvent)

Input '\$'

<https://blog.com/...>

OK Cancel

Edit Change Item

Member
NrOfRegistrations (Integer)

Type
☒ Set
 ☐ Add
 ☐ Remove

Value

\$number

Line	Column	Error

Change Object

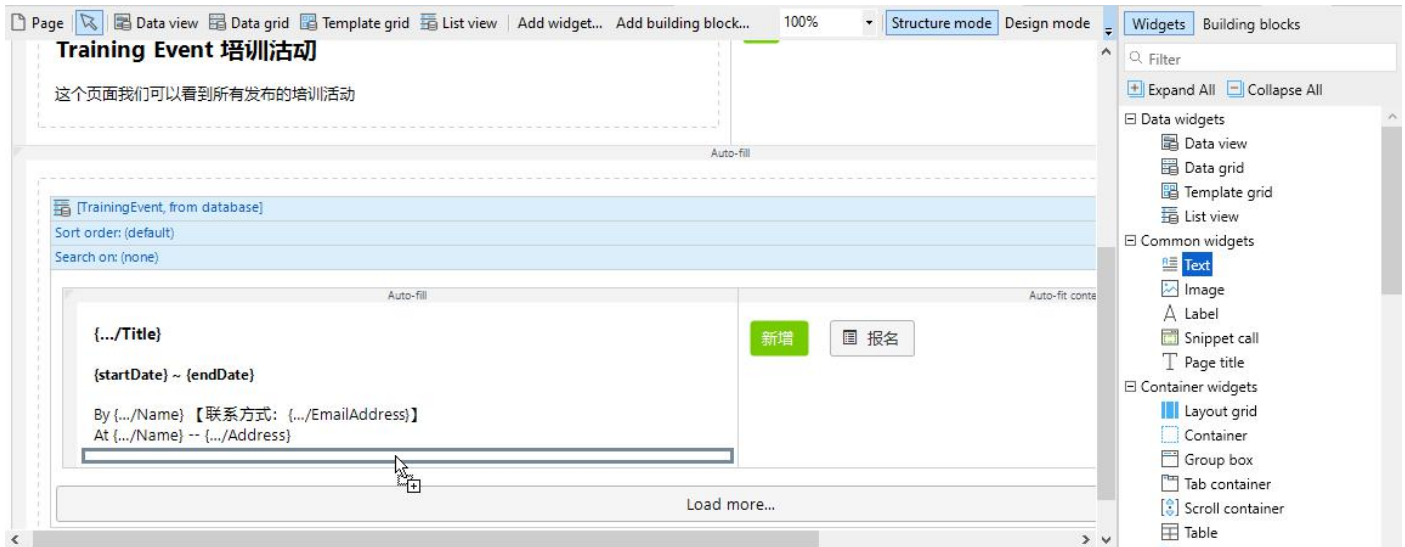
Input
Object
TrainingEvent (MyFirstModule.TrainingEvent)

Action
 Commit: ☐ Yes ☐ Yes without events ☒ No
 Refresh in client: ☐ Yes ☒ No

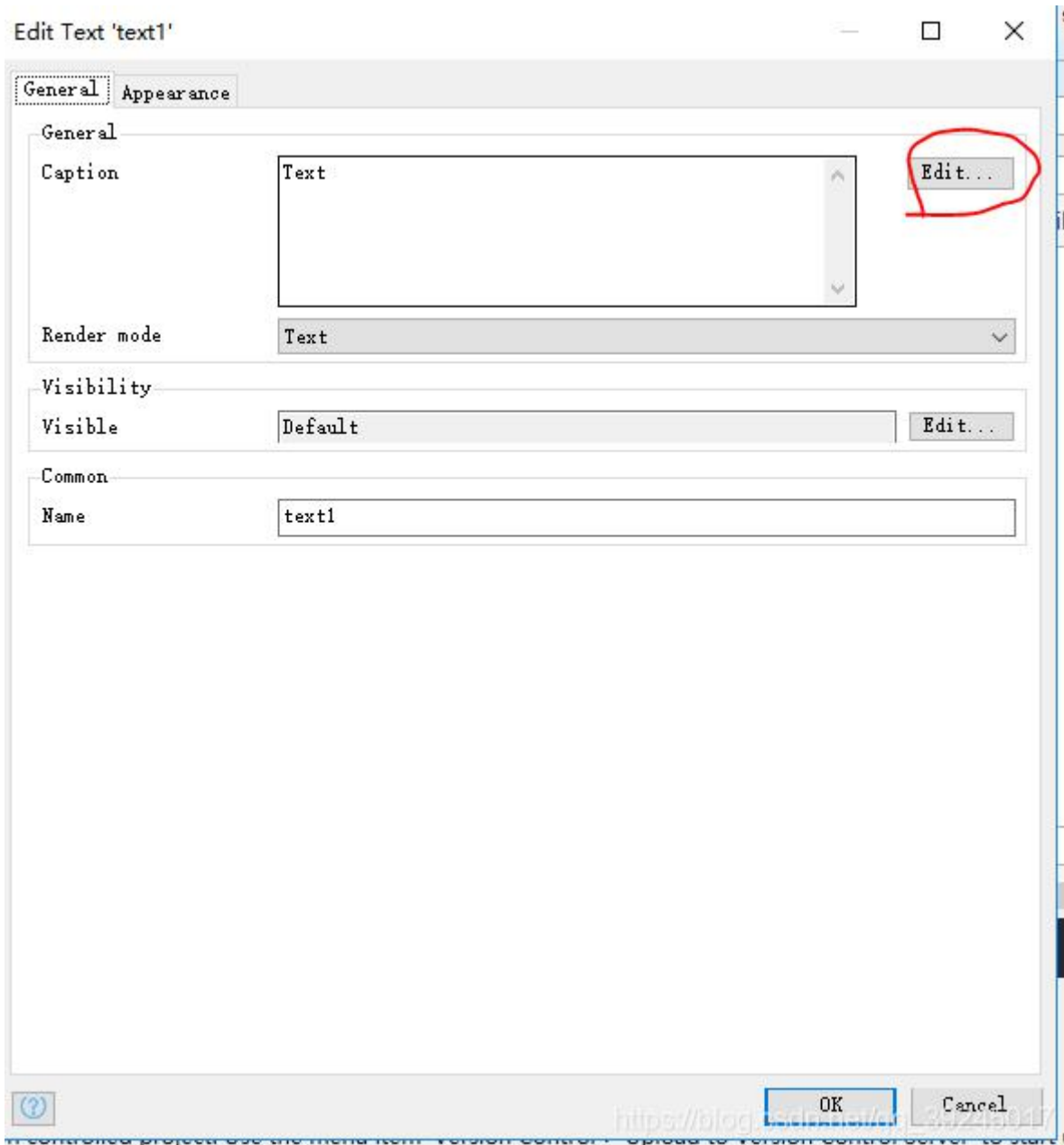
Member	Member type	Type	Value
NrOfRegistrations	Integer	Set	\$number

25. 我们基本逻辑创建好以后，现在需要在页面上展示出人员。所以需要打开我们的活动列表

TrainingEvent_Overview(双击打开)，然后在 3 号区域 Toolbox 的 Widgets 搜索 text 文本框，然后拖入到我们的页面中。然后双击拖入的文本框。



26. 在弹出的页面点击 Edit 按钮，然后选择 New 按钮。



Edit Caption

Caption

Text|

Parameters

The caption can contain parameters that are written as a number between braces, e.g. {1}.



New



Delete



Edit

▲ Move up

▼ Move down

Index	Attribute (path)	Format

Fallback text

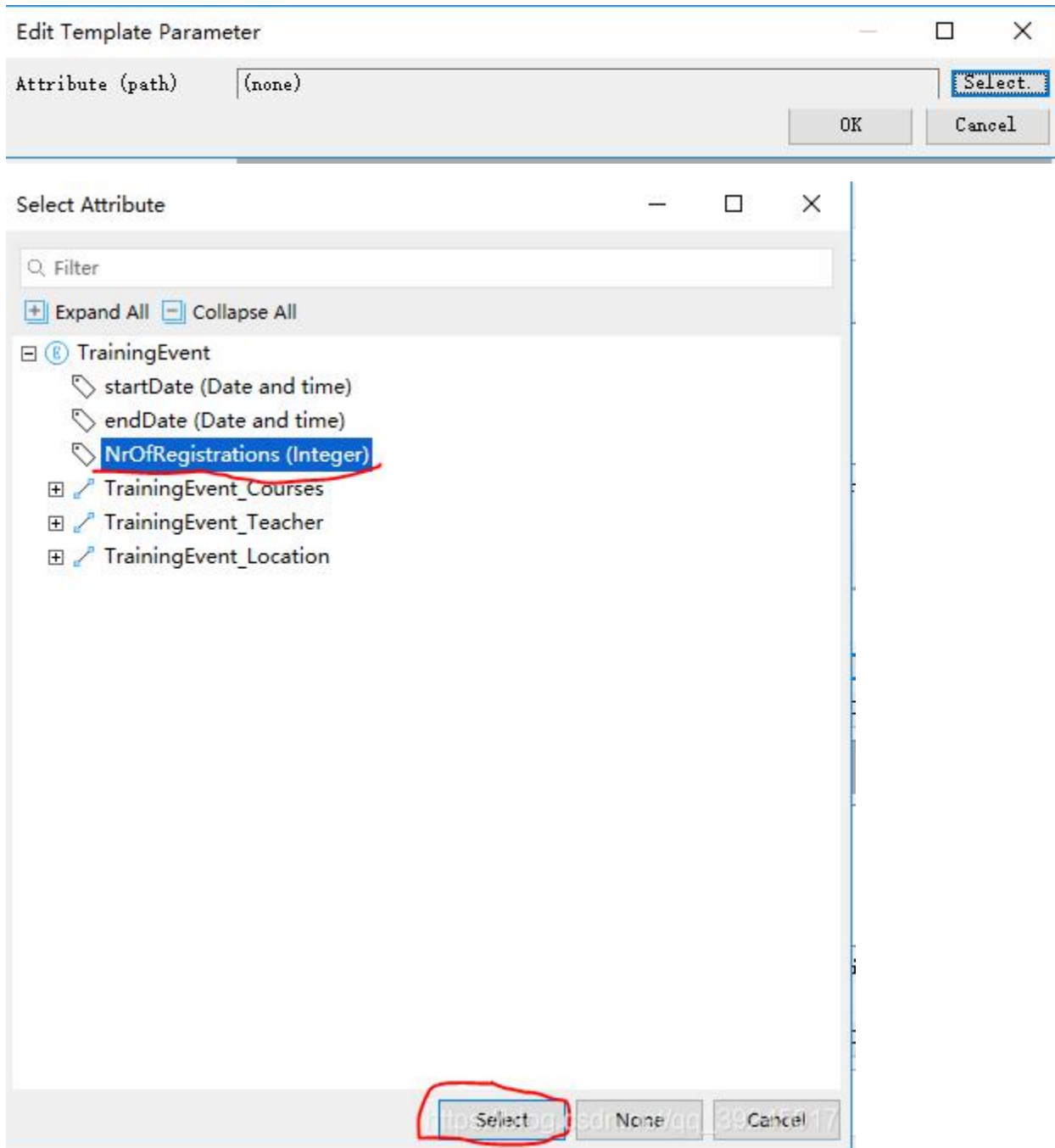
The fallback text will be displayed if there is no available data to fill the template.

OK

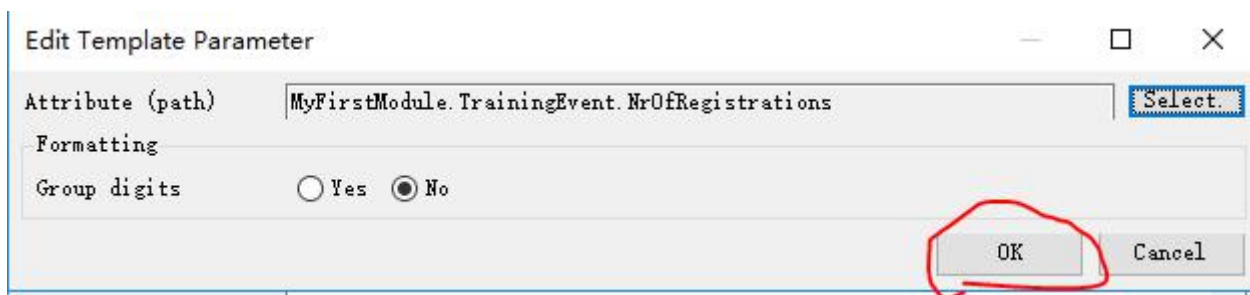
Cancel

<https://blog.mendix.com/2017/03/21/6017>

27. 然后点击 Select 按钮，选择需要展示的字段 NrOfRegistrations 人员数量信息，然后点击 Select 按钮完成选择。



28. 然后点击 Ok 按钮，然后我们在 Caption 上填写我们需要展示的信息。动态展示{1}。然后点击 OK->Ok 按钮。完成操作



Edit Caption

Caption

报名人数: {1}

Parameters

The caption can contain parameters that are written as a number between braces, e. g. {1}.

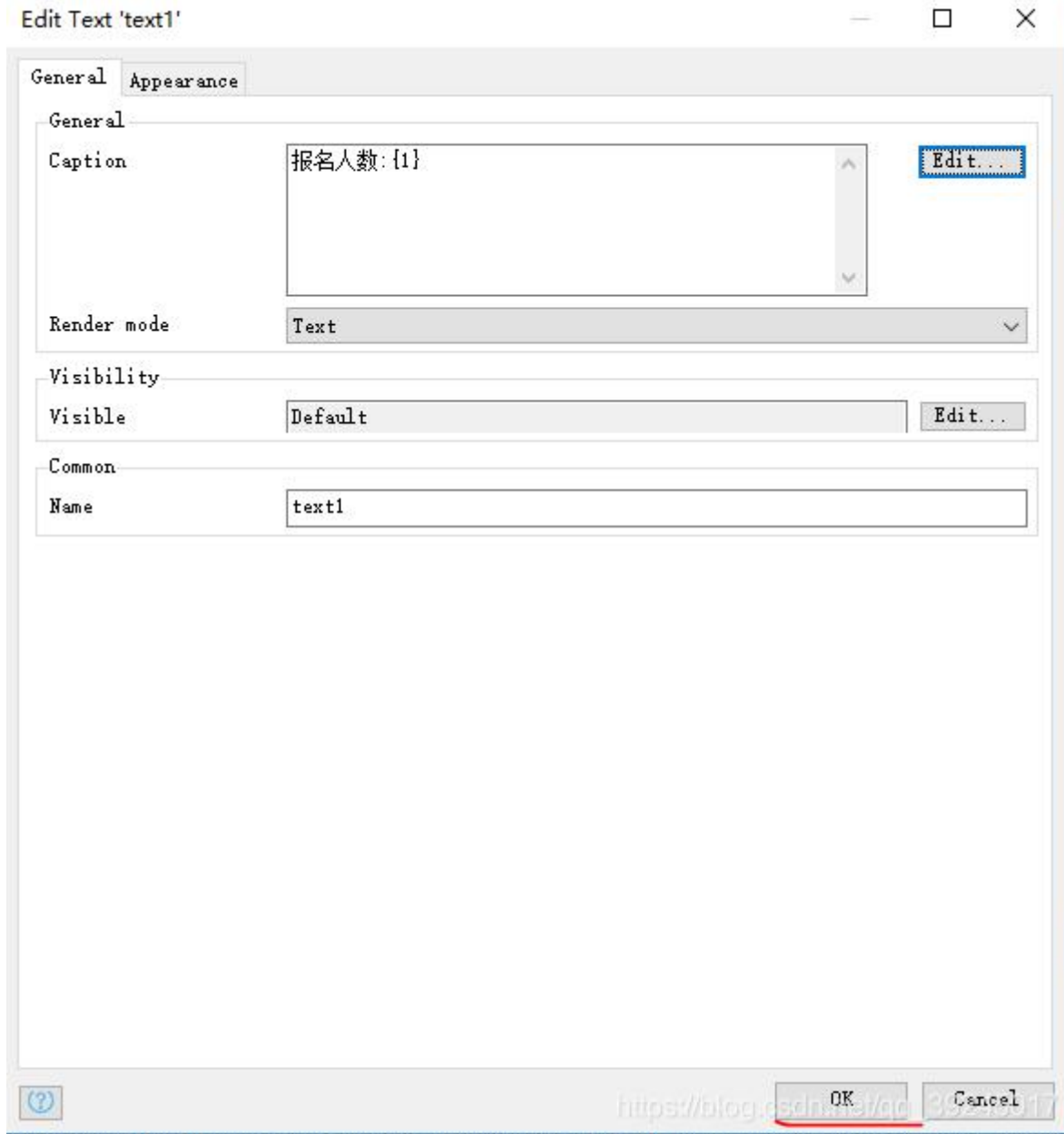
New
Delete
Edit
Move up
Move down

Index	Attribute (path)	Format
{1}	NrOfRegistrations	123456789

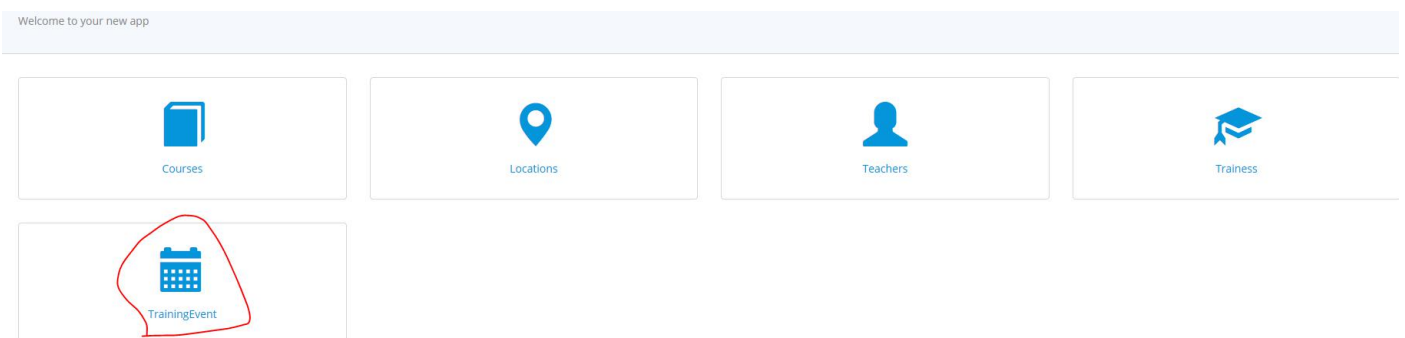
Fallback text

The fallback text will be displayed if there is no available data to fill the template.

OK
Cancel



29. 到这一步我们就算完成了，运行项目看一下效果。Ok 大功完成。



JAVA高级教程

Add

王建明
Here you can put a subtitle

>

王建明
Here you can put a subtitle

>

王建明
Here you can put a subtitle

>

https://blog.codn.net/q_39249017

JAVA高级教程

邵群超

1/14/2021

报名人数:3

Details

新增

报名

3.2.2 结束日期统计

3.2.1.1 业务需求

那现在学校又有一个需求(它怎么老是新增需求??? 心态炸裂。), 管理员觉得活动报名人员, 每次都需要自己填写结束日期, 所以它想着能不能结束日期让程序自动实现。(碍于不能得罪甲方的原则), 那么我们开始实现。

3.2.1.2. 需求分析

根据分析我们可以在用户报名时选择课程信息(包含培训天数)以后和选择开始日期, 触发我们微流, 需要微流做什么事呢? 就是让他将开始日期自动加上培训天数得出结束日期, 然后返回到页面。

3.2.1.3. 项目实施

分析完毕, 我们开始实现。

1. 首先双击我们目录树中的 TrainingEvent_NewEdit(活动新增)页面, 然后双击我们课程框(Courses), 在弹出的页面选择 On change 事件为 Call a microflow , 触发微流事件。在弹出的页面选择 New 按钮, 重新创建一个 Microflow 微流。

coursemanager - Mendix Studio Pro 8.12.4

File Edit View Project Run Version Control Language Help Run Locally View

Project Explorer

- Project 'coursemanager'
- System
- MyFirstModule
 - Domain Model
 - ACO_Ade_Registration_SetNubOfRegi
 - ACT_TrainingEvent_New
 - Courses_NewEdit
 - Courses_Overview
 - Home_Web
 - Images
 - Location_NewEdit
 - Locations_Overview
 - Registration_NewEdit
 - Teacher_NewEdit
 - Teachers_Overview
 - Trainee_NewEdit
 - Trainess_Overview
 - TrainingEvent_NewEdit**
 - TrainingEvent_Overview
 - TrainingEvent_View

TrainingEvent_NewEdit (MyFirstModule)

PopupLayout > TrainingEvent_NewEdit

[TrainingEvent_page parameter]

Courses

[TrainingEvent_Courses/Courses/Title]

start date

[startDate]

end date

[endDate]

Teacher

[TrainingEvent_Teacher/Teacher/Name]

Location

[TrainingEvent_Location/Location/Name]

Console

Start Stop Clear Scroll Lock Advanced

Date/time	Log node	Message
2021-01-15 17:39:52...	Core	Initialized license.
2021-01-15 17:39:53...	Core	Mendix Runtime successfully started, the application is now available.
2021-01-15 20:49:54...	Core	Maximum run time exceeded, framework is now terminating
2021-01-15 20:49:54...	Core	Mendix Runtime is shutting down now...
2021-01-15 20:49:54...		(Logging disconnected)

Stories Changes Errors (0) Console Debugger Variables

Edit Reference Selector 'referenceSelector'

General Common Appearance Selectable objects

General

Select using ☐ Page ☒ Drop-down

Empty option caption

Go-to page (none)

Page title ☐ Override page t

Data source

Attribute (path) MyFirstModule.TrainingEvent_Courses/MyFirstMo Select

Label

Show Label ☒ Yes ☐ No

Label option Courses Ed

Editability

Editable ☒ Default ☐ Never ☐ Conditionally

Read-only style ☒ Based on data view (Control) ☐ Control ☐ Text

Visibility

Visible Default Ed

Validation

Type (none)

Events

On change

Do nothing

Do nothing

Show a page

Call a microflow

Call a microflow

Open link

Create object

Save changes

Cancel changes

Close page

Delete

Synchronize

Sign out

Reference set selector

Input reference set selec

File widgets

File manager

Image uploader

Image viewer

Properties Toolbox Co

Select Microflow

Filter

Expand All Collapse All

App Store modules

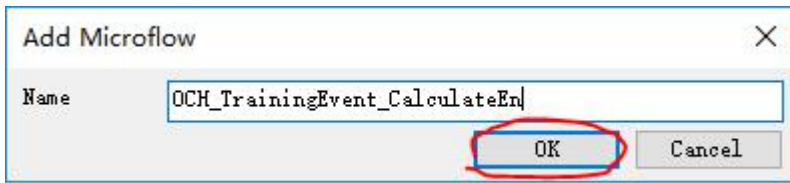
System

MyFirstModule

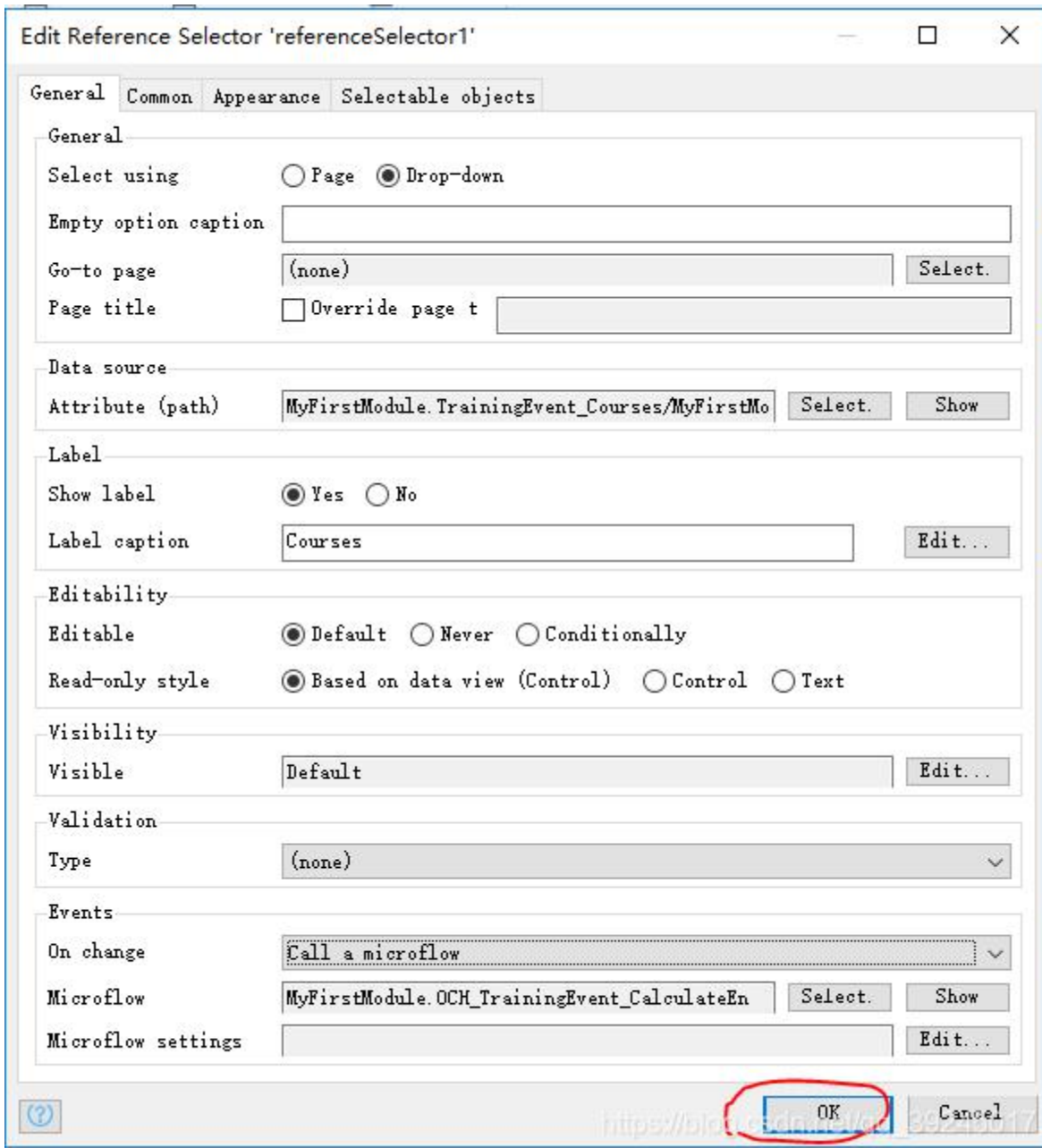
- ACO_Ade_Registration_SetNubOfRegi
- ACT_TrainingEvent_New

Select New None Cancel

- 在弹出的页面输入 Microflow 微流名称 OCH_TrainingEvent_CalculateEn（命名规范），然后点击 Ok->Ok 按钮完成新建。

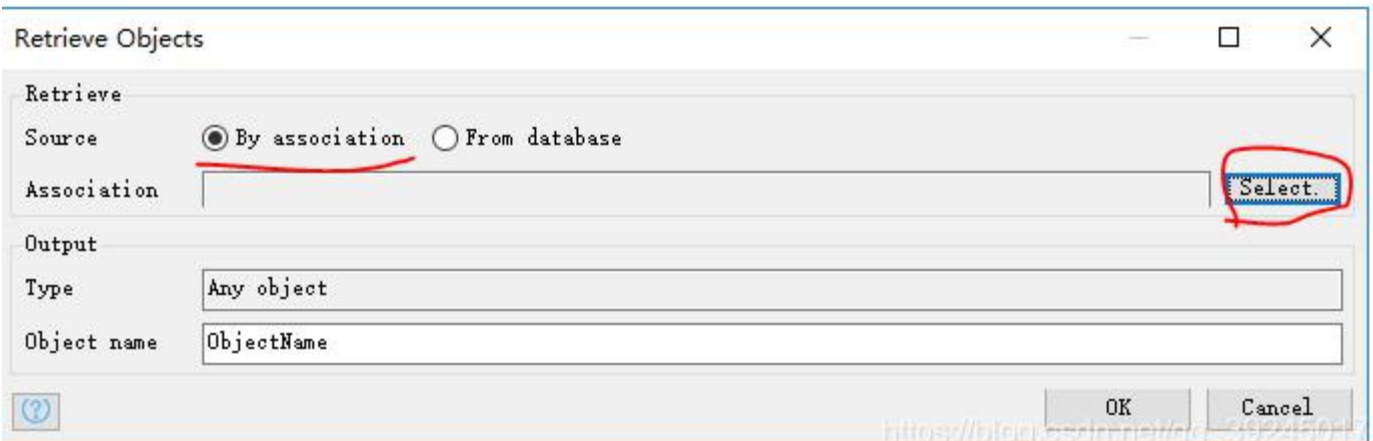
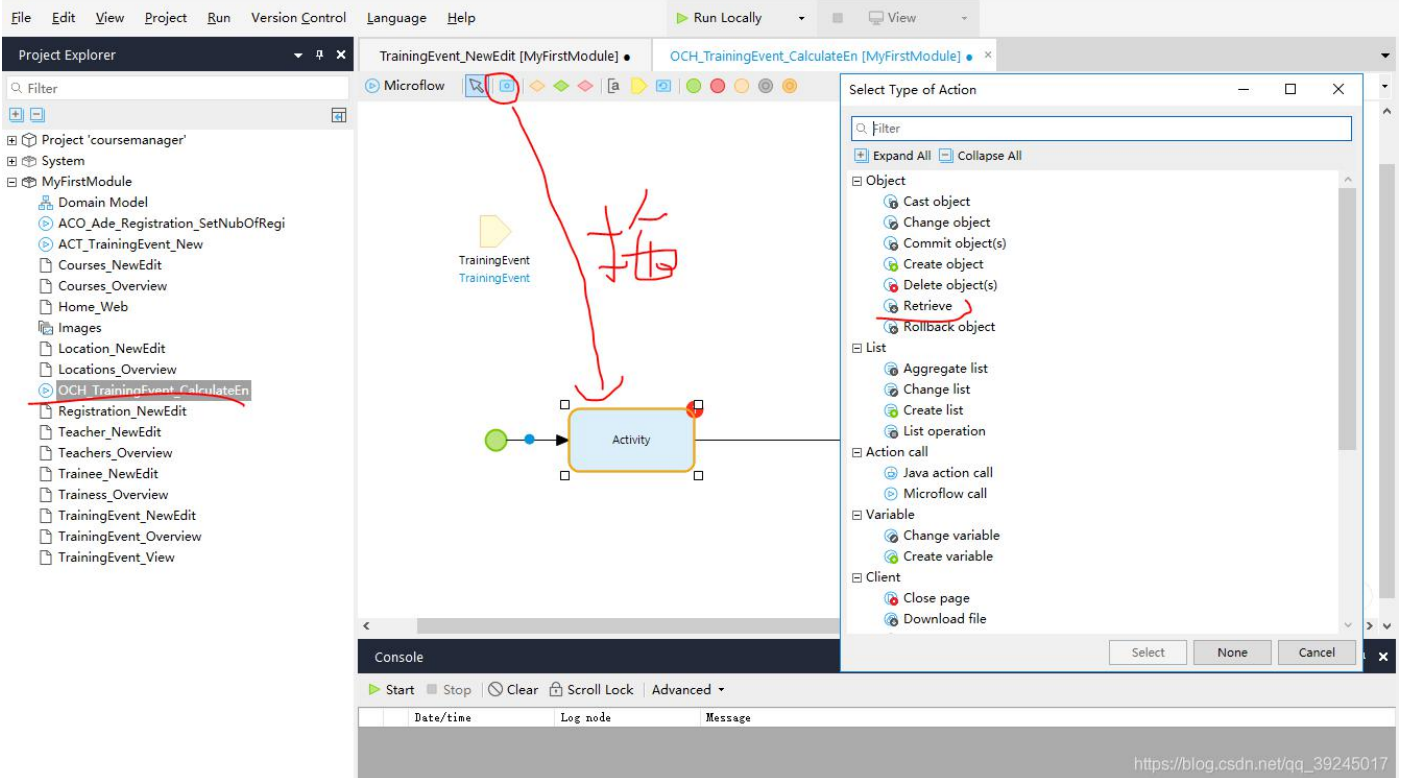


Dialog box titled "Add Microflow". The "Name" field contains "OCH_TrainingEvent_CalculateEn". The "OK" button is highlighted with a red circle.

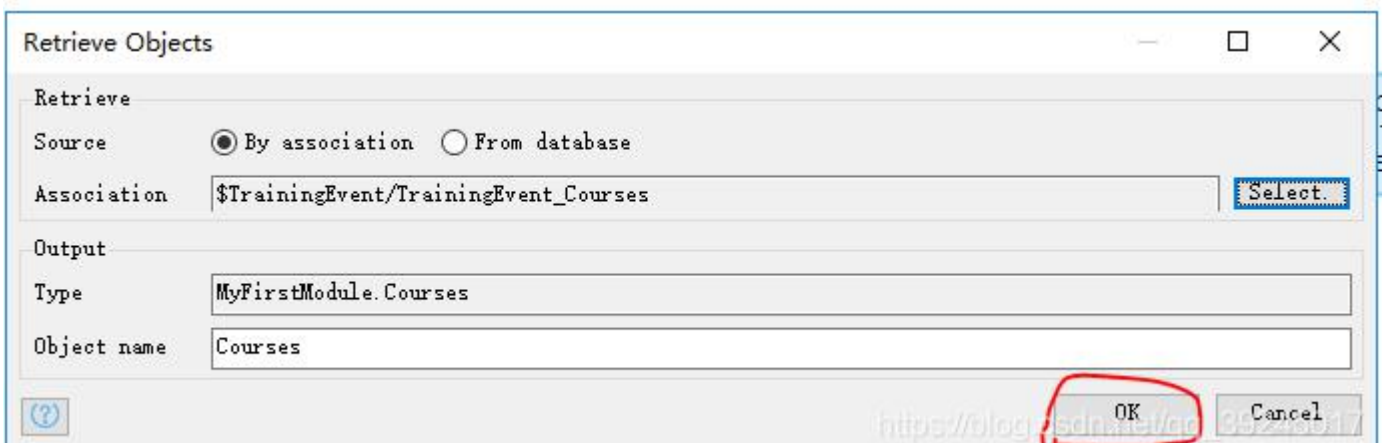
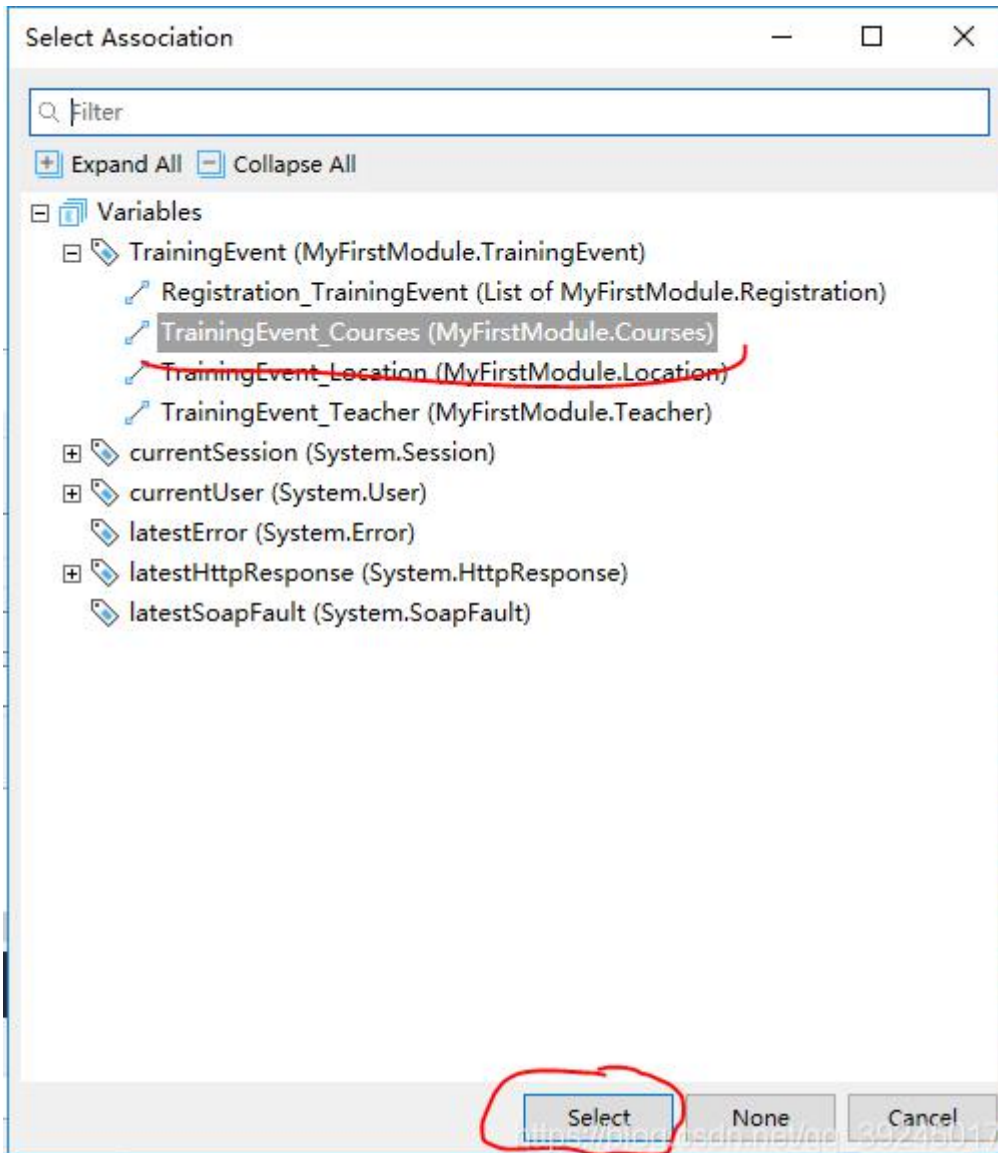


Dialog box titled "Edit Reference Selector 'referenceSelector1'". The "General" tab is selected. The "Select using" section has "Drop-down" selected. The "Data source" section has "Attribute (path)" set to "MyFirstModule.TrainingEvent_Courses/MyFirstMo". The "Label" section has "Show label" set to "Yes" and "Label caption" set to "Courses". The "Editability" section has "Editable" set to "Default" and "Read-only style" set to "Based on data view (Control)". The "Visibility" section has "Visible" set to "Default". The "Validation" section has "Type" set to "(none)". The "Events" section has "On change" set to "Call a microflow" and "Microflow" set to "MyFirstModule.OCH_TrainingEvent_CalculateEn". The "OK" button is highlighted with a red circle.

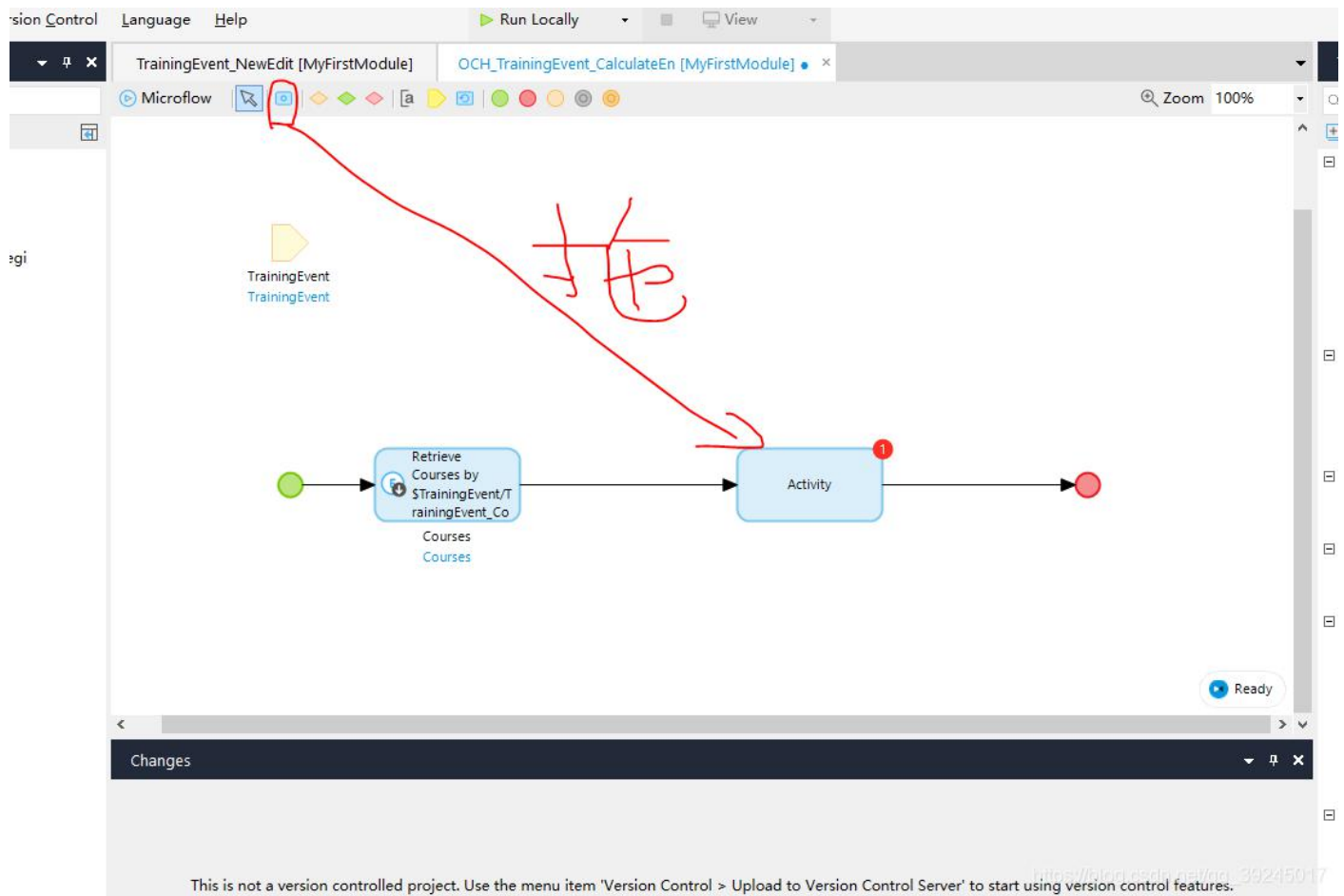
- 创建完毕以后，我们双击目录树中刚创建的 Microflow 微流 OCH_TrainingEvent_CalculateEn，然后在工作台拖入一个 Activity 事件，并双击，选择 Retrieve 事件(用来获取课程信息中的培训时间用来做最后的计算)。在弹出的页面选择 By association（从内存中获取），然后点击 Select 按钮。



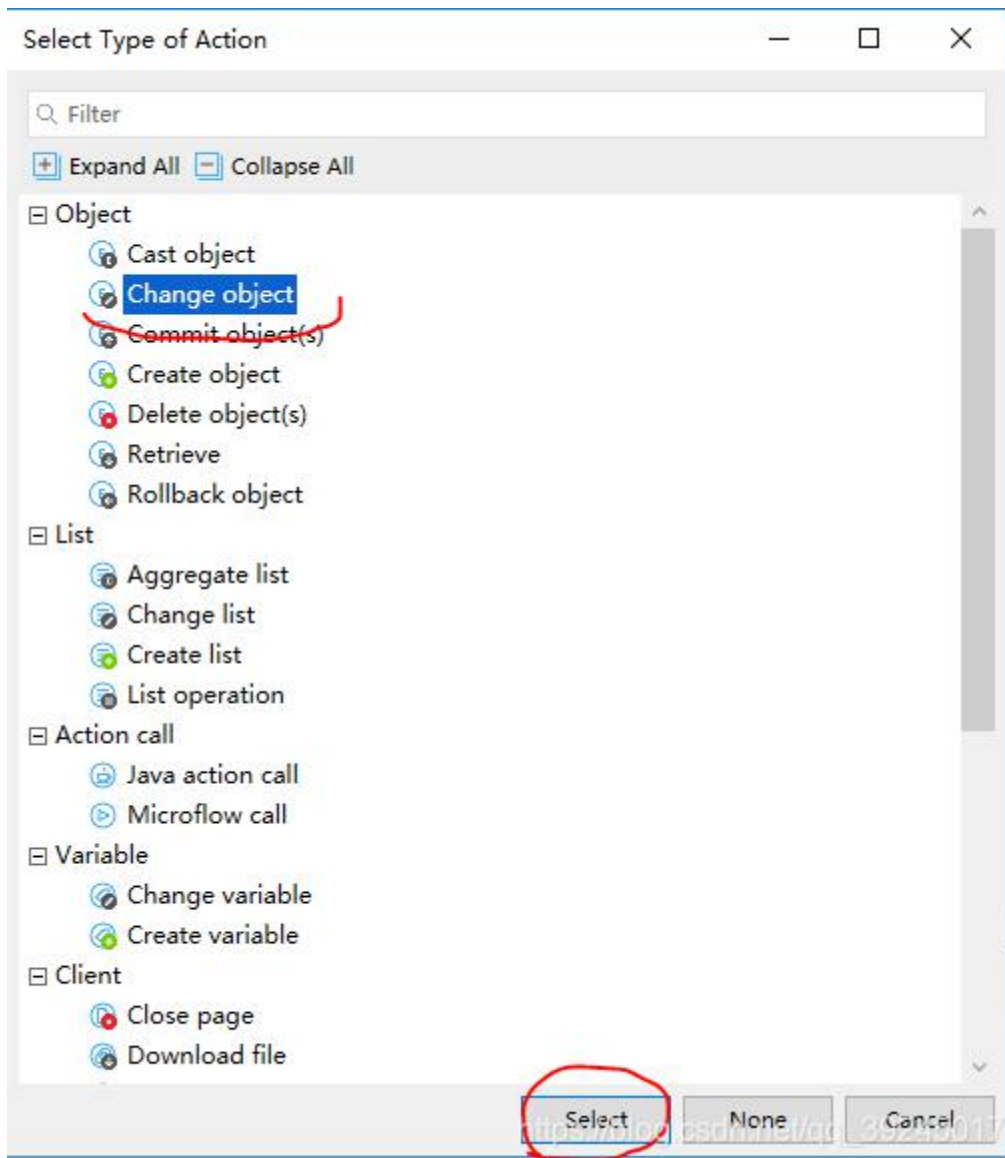
4. 在弹出的页面我们选择与活动关联的课程信息(Courses)数据模型，然后点击 Select->Ok 按钮，完成选择。



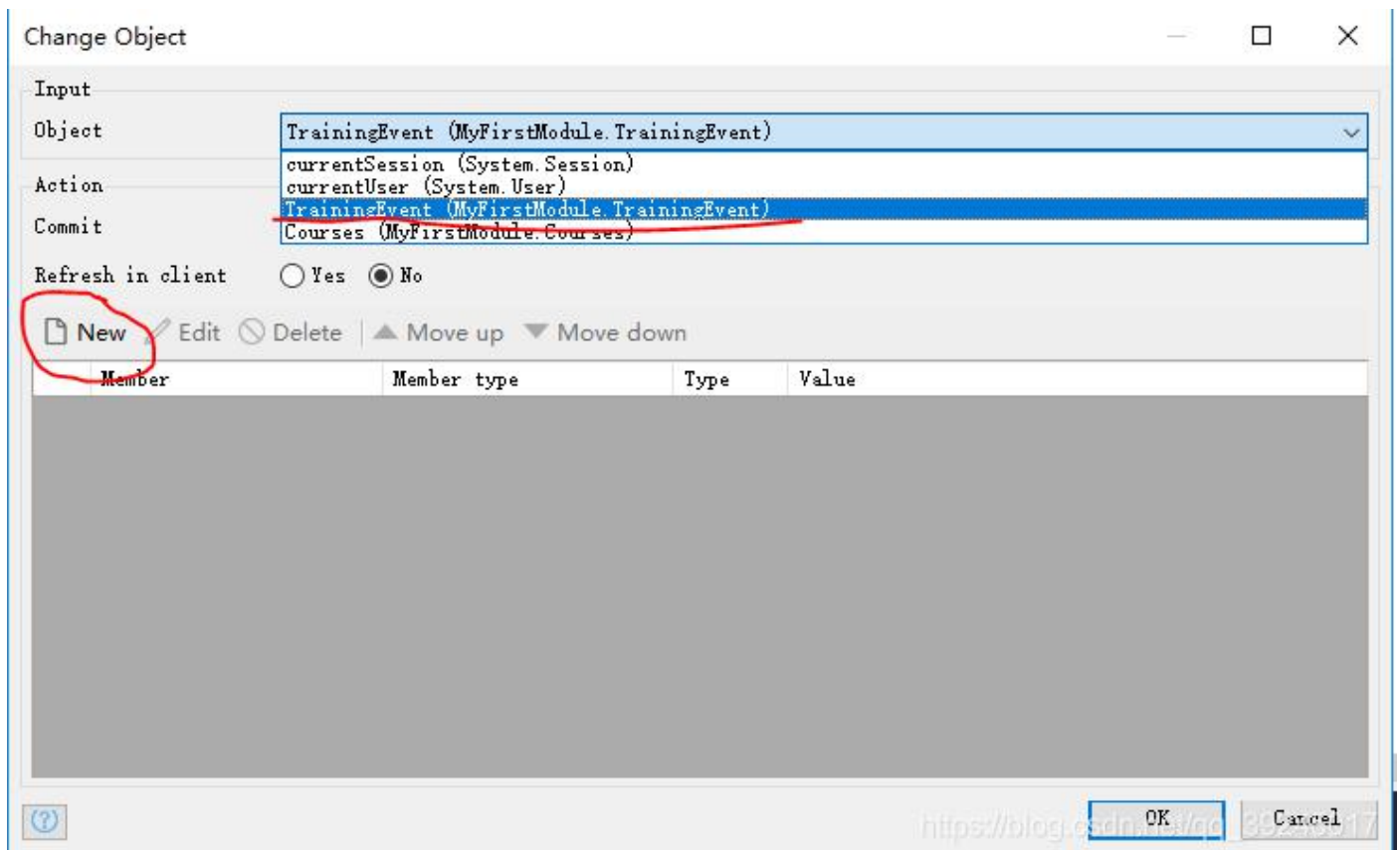
5. 我们再拖入一个 Activity 事件，用来更改计算后的 endDate(结束时间)。双击拖入的 Activity 事件。



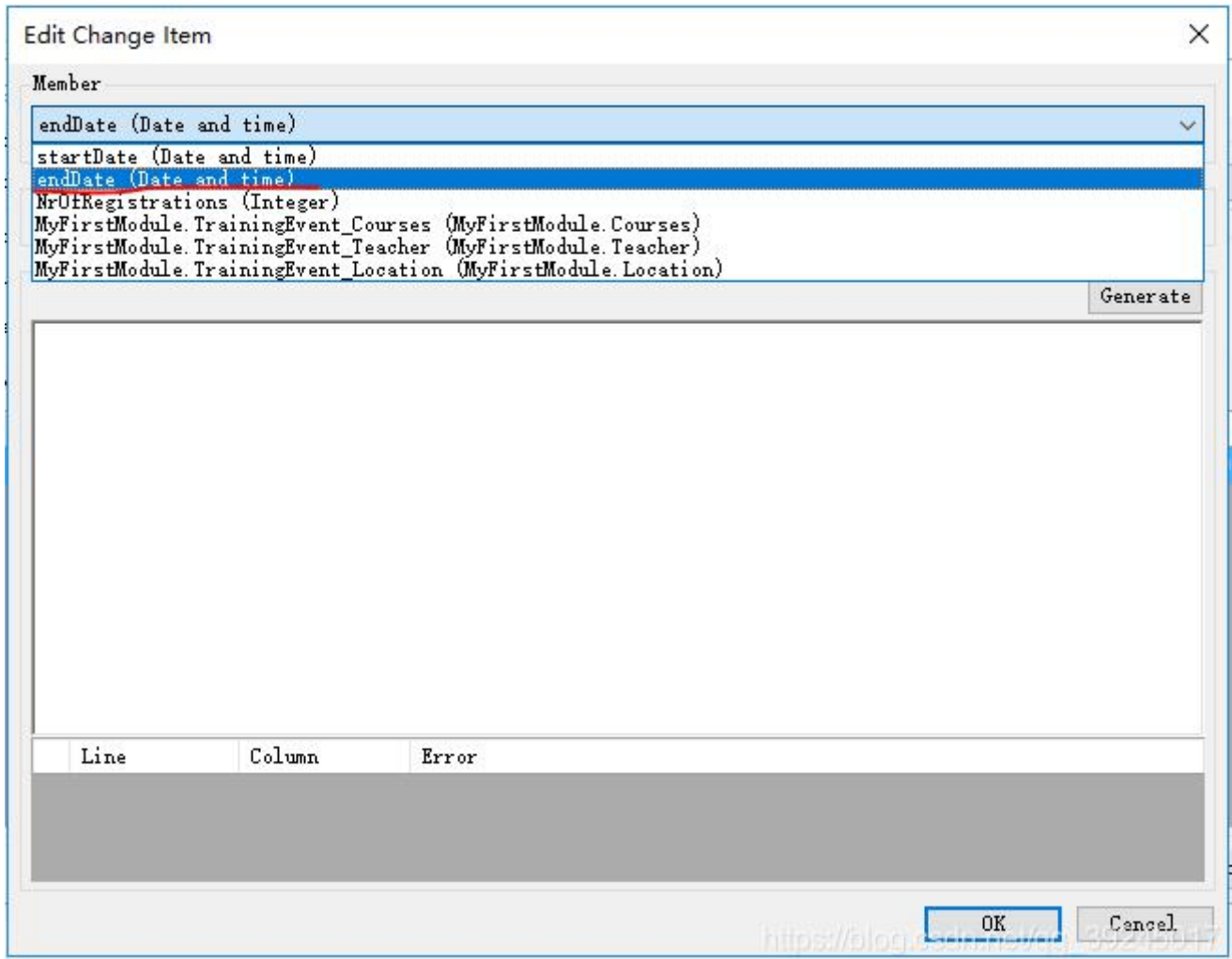
6. 选择 Change Object 事件。然后点击 Select 按钮。完成事件选择。



7. 在弹出的页面我们选择 Object 为 TrainingEvent(活动), 用来改活动里的 endDate (结束时间)。然后点击 New 按钮(新建我们的规则)。



8. 首先需要选择要修改的活动 endDate(结束日期)。



9. 在 Value 里边来编写我们的规则。首先用到的函数是 addDays(追加时间), 可参考常用的函数[点击跳转](#), 然后我们输入我们的函数, 在 2 部分先获取到活动的开始日期, 然后在 3 部分选择活动所需天数, 系统就可以自动追加天数了。然后点击 Ok->Ok 按钮, 完成对规则的编写(自己写一遍, 不建议复制)。

addDays(\$TrainingEvent/startDate,\$Courses/Duration)

Edit Change Item

Member
endDate (Date and time)

Type
☒ Set ☐ Add ☐ Remove

Value
Generate

```
addDays($TrainingEvent/startDate,$Courses/Duration)
```

1 2 3

Line	Column	Error
------	--------	-------

OK Cancel

Change Object

Input
Object
TrainingEvent (MyFirstModule.TrainingEvent)

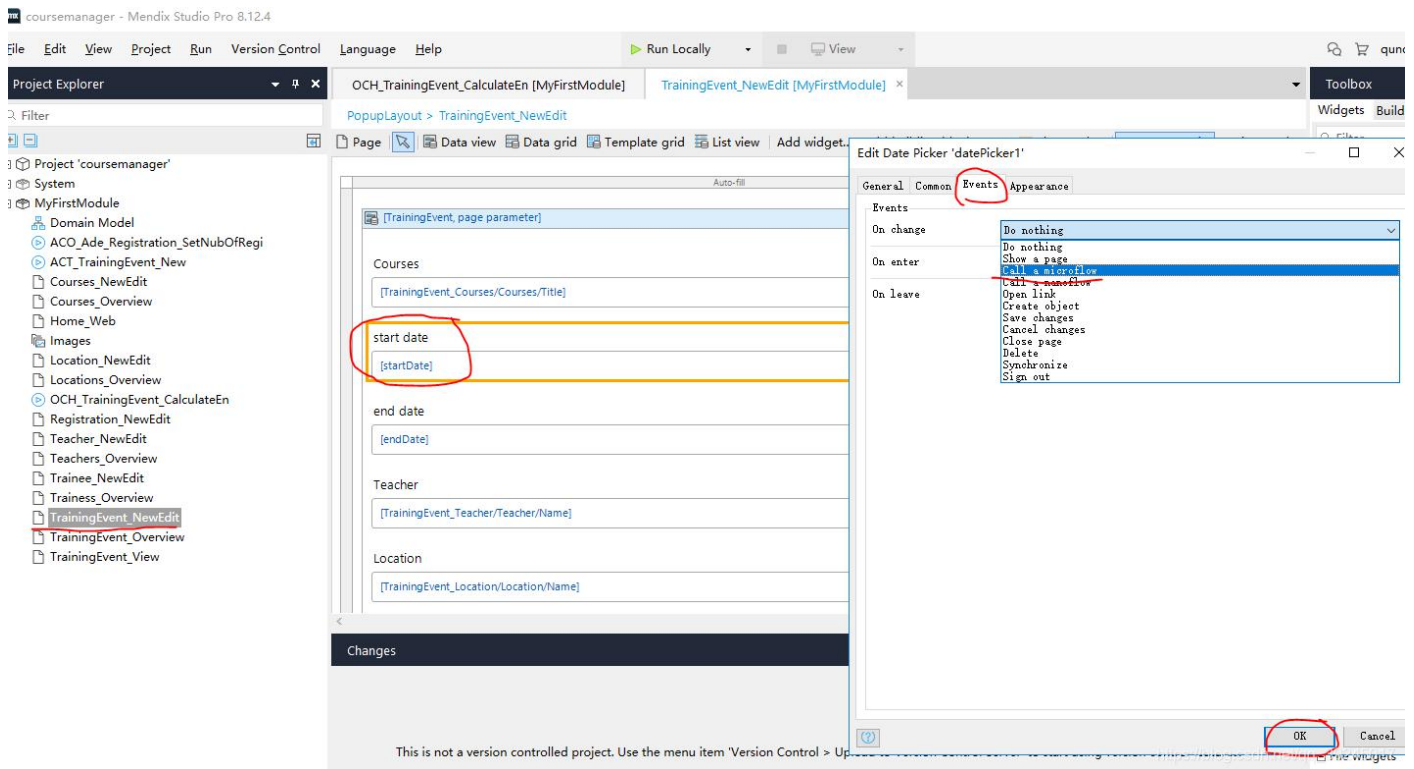
Action
Commit ☐ Yes ☐ Yes without events ☒ No
Refresh in client ☐ Yes ☒ No

New Edit Delete Move up Move down

Member	Member type	Type	Value
endDate	Date and time	Set	addDays(\$TrainingEvent/startDate,\$Courses/Duration)

OK Cancel

10. 同样的我们需要在开始时间选择时调用我们的微流。所以双击目录树的 TrainingEvent_NewEdit 新增页面，然后双击 start date 开始日期，然后在弹出的页面选择 Events 事件的 On change 事件为 Call a microflow。



11. 然后选择我们刚才创建的微流 OCH_TrainingEvent_CalculateEn,然后选择 Select->Ok 按钮,完成该部分的设计。

Select Microflow



Filter

Expand All Collapse All

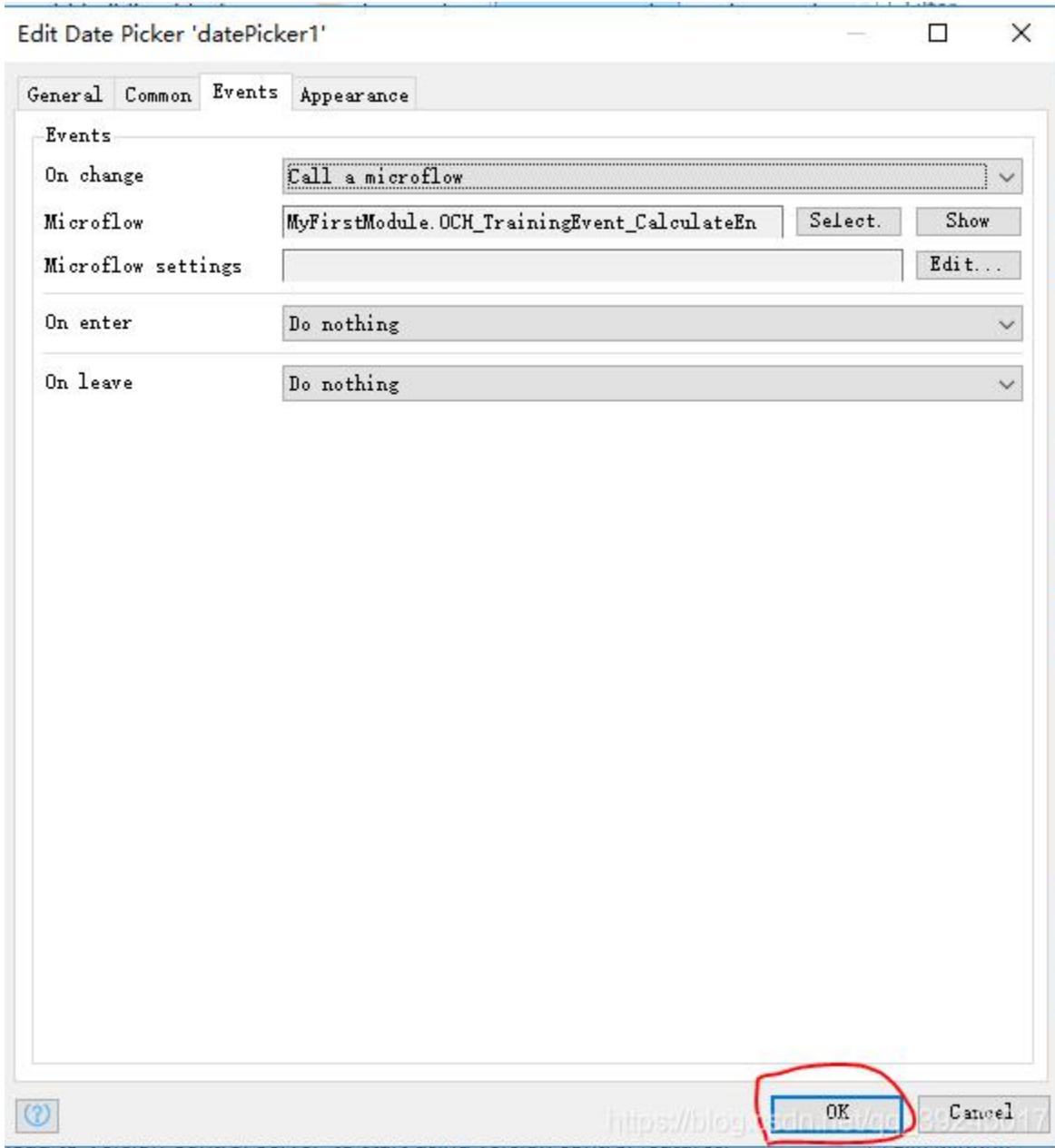
- App Store modules
- System
- MyFirstModule
 - ACO_Ade_Registration_SetNubOfRegi
 - ACT_TrainingEvent_New
 - OCH_TrainingEvent_CalculateEn**

Select

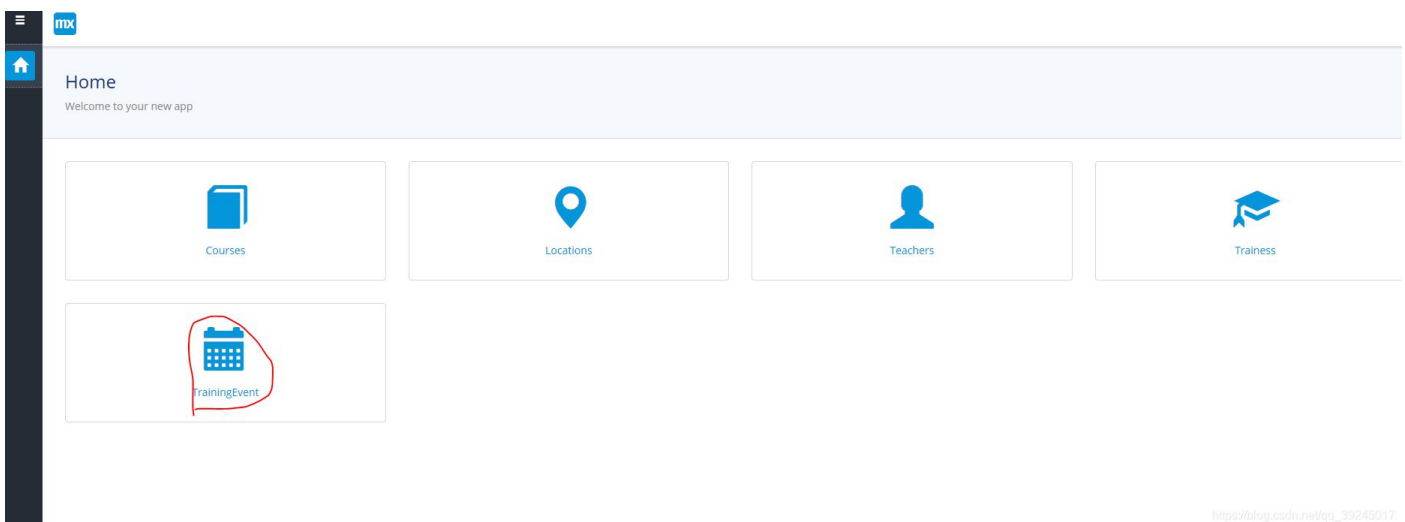
New

None

Cancel



12. 我们运行项目查看一下效果，我们可以看到结束日期自动计算出来了。



https://blog.csdn.net/qz_35245017

	JAVA高级教程 邵群超 1/14/2021 报名人数:0	Details 新增 报名
	JAVA高级教程 邵群超 1/14/2021 报名人数:0	Details 新增 报名

https://blog.csdn.net/qq_39245017

Edit Training Event

Courses

JAVA高级教程

start date

1/18/2021

end date

1/28/2021

Teacher

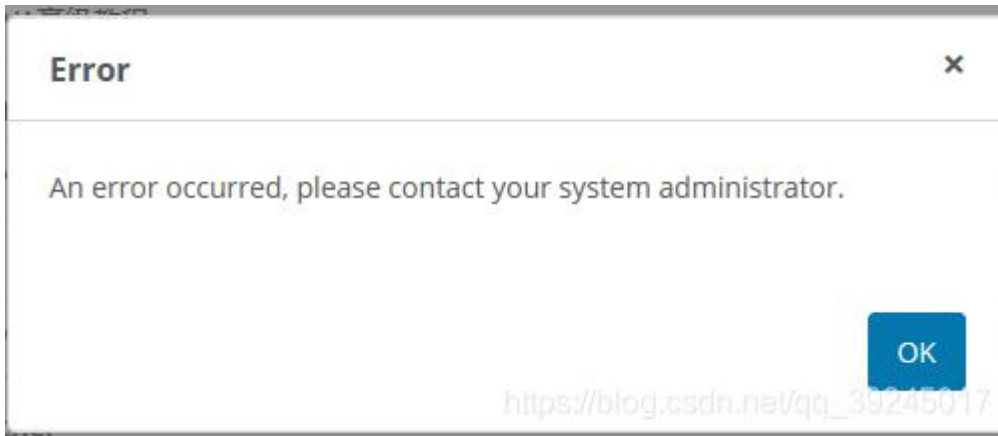
Location

Save

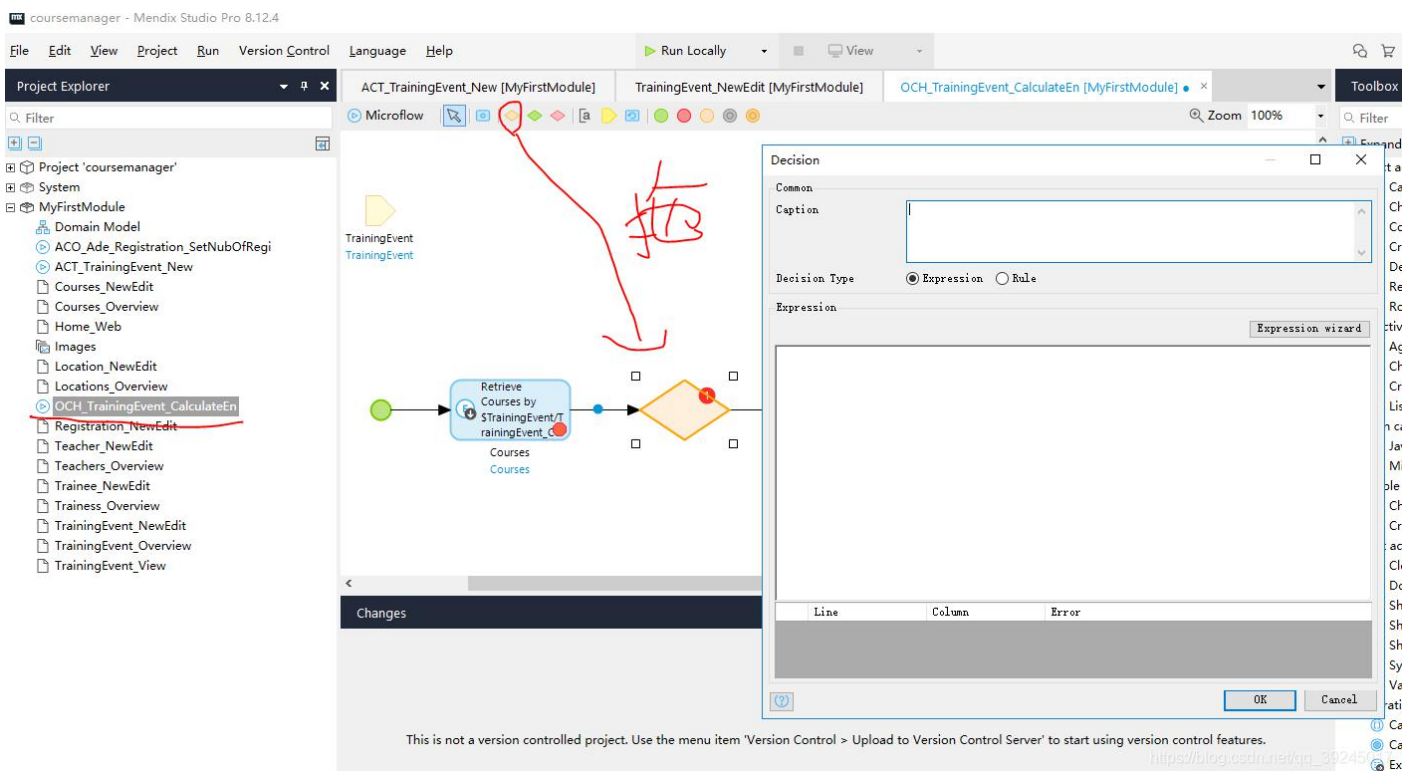
Cancel

https://blog.csdn.net/qq_39245017

13. 经过演示我们发现虽然实现了效果，但是老是会有一个提示弹框(出现错误，请与系统管理员联系。)，说我们有一个错误。这时候我们可以 DeBug 进行问题跟踪，我写到章节[二、DeBug 问题跟踪](#)，可以点击看怎么进行的跟踪。



14. 问题跟踪完成以后，我们做一下非空的处理逻辑，双击我们的微流，然后拖入一个判断分支，并双击打开它。



15. 然后在 Expression 中判断我们的逻辑，输入**\$选择 **TrainingEvent**(活动数据类型)，然后输入/**选择 startDate，不等于!= empty。判断逻辑做好以后，我们在 Caption 起一个名称，可以随便起，但是要有意义。我这里就 startDate not empty。然后点击 Ok 按钮(不建议复制)。

\$TrainingEvent/startDate != empty

Decision

Common

Caption startDate not empty

Decision Type ☒ Expression ☐ Rule

Expression

Expression wizard

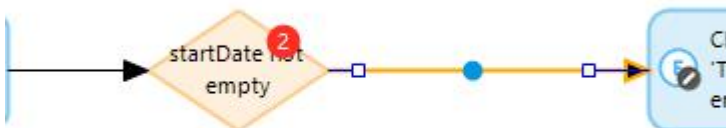
`$TrainingEvent/startDate != empty`

Line	Column	Error

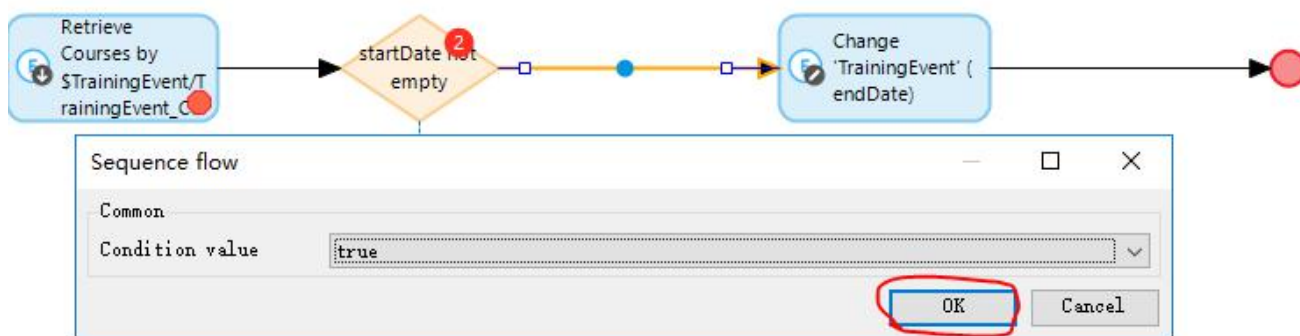
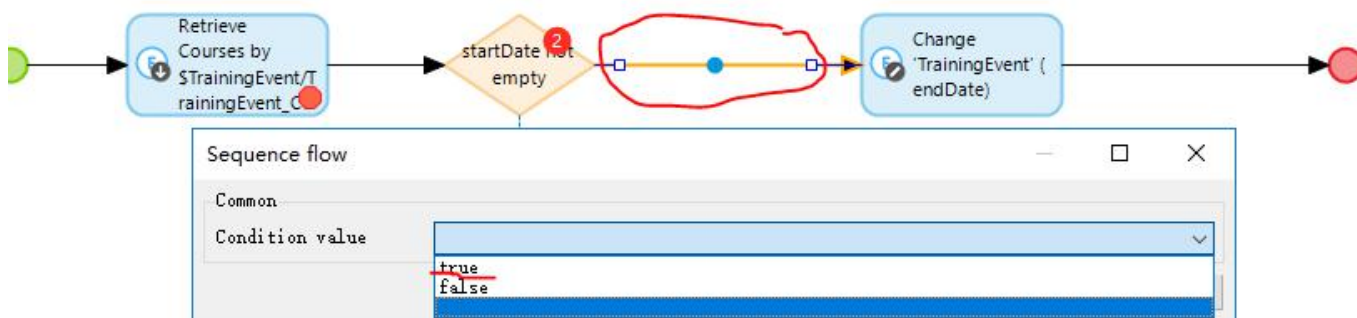
OK Cancel

the menu item 'Version Control > Upload to Version Control Server' to start using version control features.

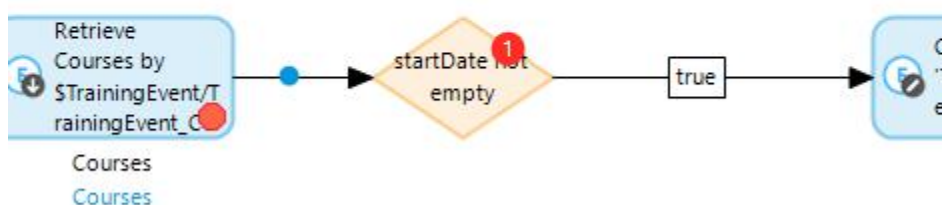
16. 这时候发现工作台报了两个错误，是因为我们需要给分支判断结束以后的走向。



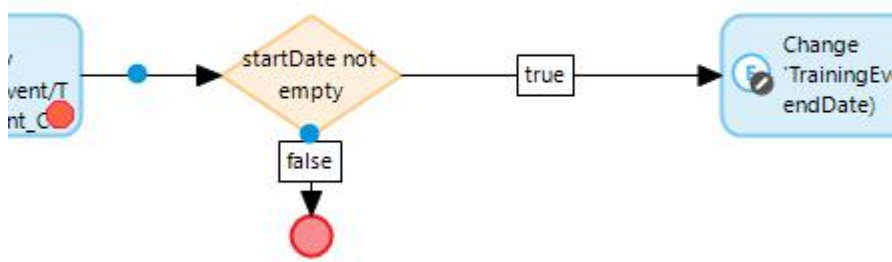
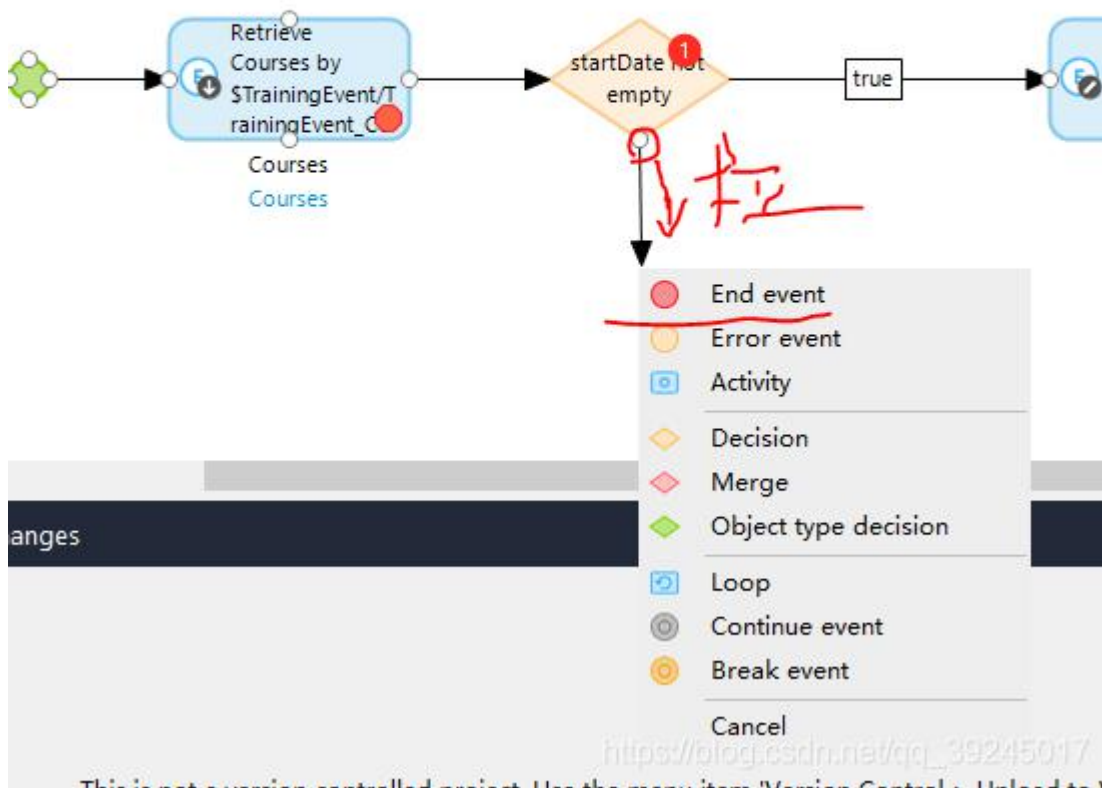
17. 我们双击下图中红框内容，然后给他选择判断为 true 的走向为继续执行(不等于空的时候继续执行)，然后点击 Ok 按钮。



18. 这个时候还有一个错误，需要给他判断为 false 的时候一个走向(如果等于空就让他结束)。



19. 然后按照下图所示，鼠标左键，直接拉。拉好适当位置以后松开，他就会让你选择事件。这时候我们选择 End event(结束事件)，就可以了。



20. 同样的 我们也要判断一下课程中所需时长为空也不触发计算的逻辑。这里就不演示了，跟上边步骤一样(不建议复制)。\$Courses/Duration != empty

Decision

Common

Caption: Duration not empty

Decision Type: ☒ Expression ☐ Rule

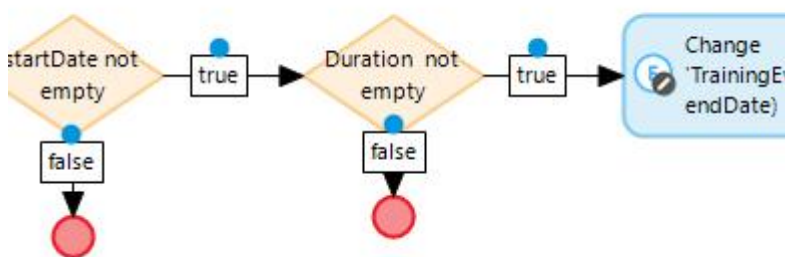
Expression

Expression wizard

```
$Courses/Duration != empty
```

Line	Column	Error

OK Cancel



21. 设置完逻辑以后，我们点击保存运行。我们可以发现这时候就没有刚才的报错提示框了。该需求处理结束。

3.3 举例 3（用微流做表单验证）

3.3.1 业务需求

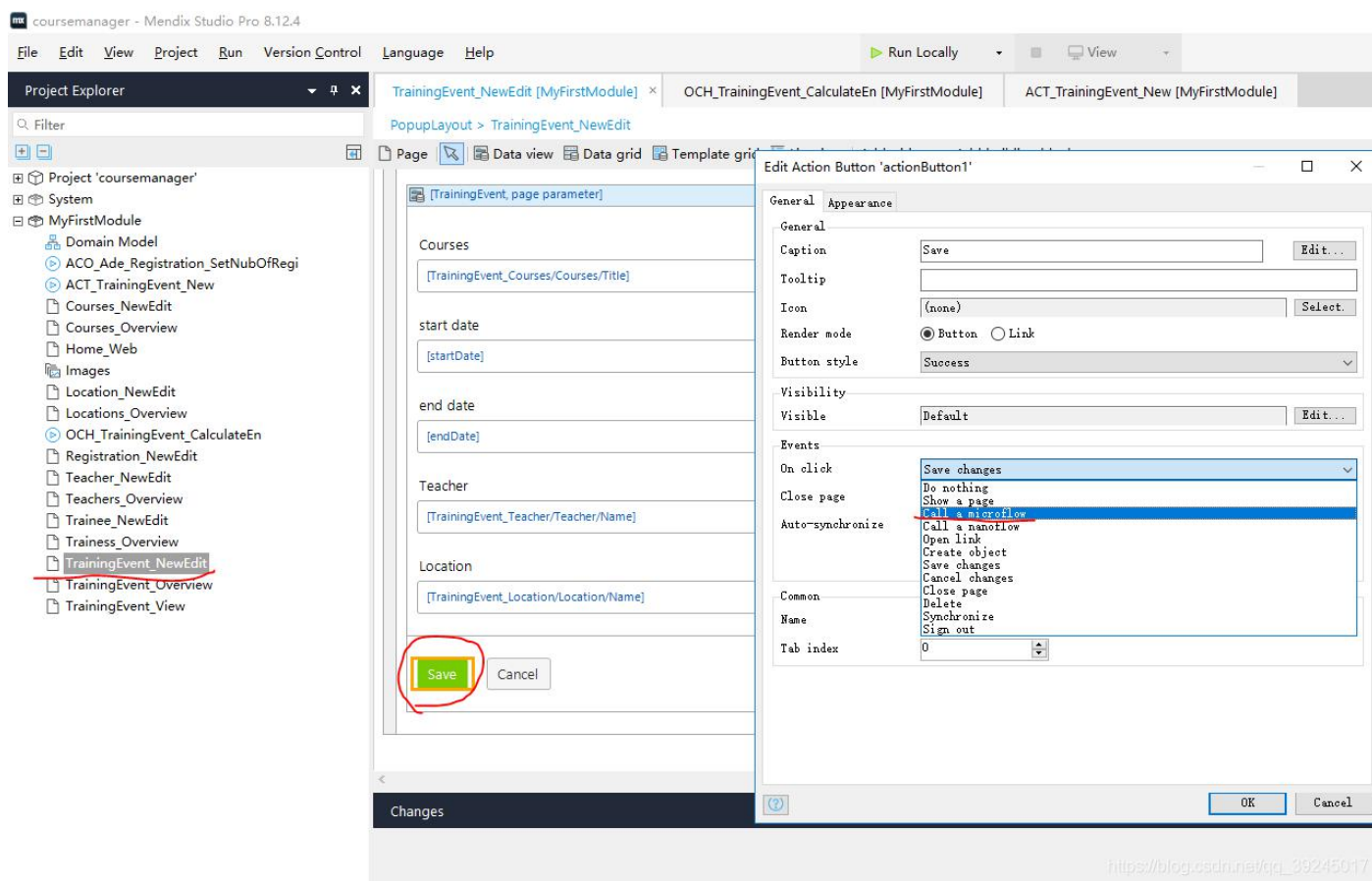
现在学校又有一个需求(心态炸裂。???)，管理员觉得新增活动时，空的数据还可以保存(说程序逻辑不对)，让我们修改为空的数据程序不保存并且提示错误信息。(没办法啊)，我们只能老老实实的实现功能。

3.3.2 需求分析

根据分析我们需要在活动新建提交保存的时候，触发微流让微流做一下非空的判断，然后阻止表单提交。

3.3.3 项目实现

1. 我们双击 TrainingEvent_NewEdit 活动新增页面，然后双击 Save 按钮，在弹出的窗口 On click 选择 Call a microflow，创建一个微流。



2. 我们选择 New 新建一个，在弹出框中输入微流名称 ACT_TrainingEvent_Save(命名规范)，然后点击 Ok-Ok 按钮，完成创建。

Select Microflow

Filter

- ☐ App Store modules
- ☐ System
- ☒ **MyFirstModule**
 - ☐ ACO_Ade_Registration_SetNubOfRegi
 - ☐ ACT_TrainingEvent_New
 - ☐ OCH_TrainingEvent_CalculateEn

Add Microflow

Name

Edit Action Button 'actionButton1'

General Appearance

General

Caption Save Edit...

Tooltip

Icon (none) Select.

Render mode ☒ Button ☐ Link

Button style Success

Visibility

Visible Default Edit...

Events

On click Call a microflow

Microflow MyFirstModule.ACT_TrainingEvent_Save Select. Show

Microflow settings Edit...

Disabled during action ☒ Yes ☐ No

Common

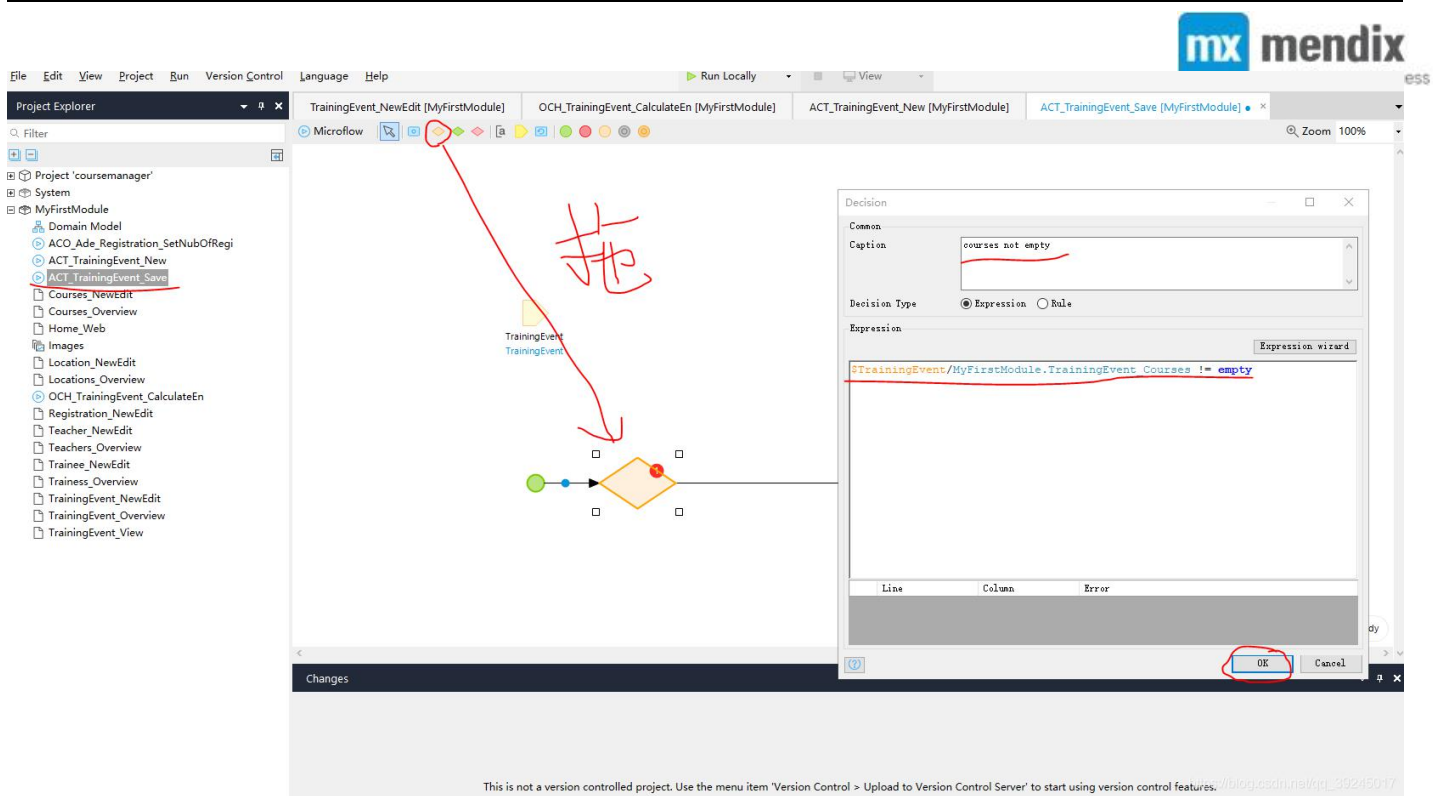
Name actionButton1

Tab index 0

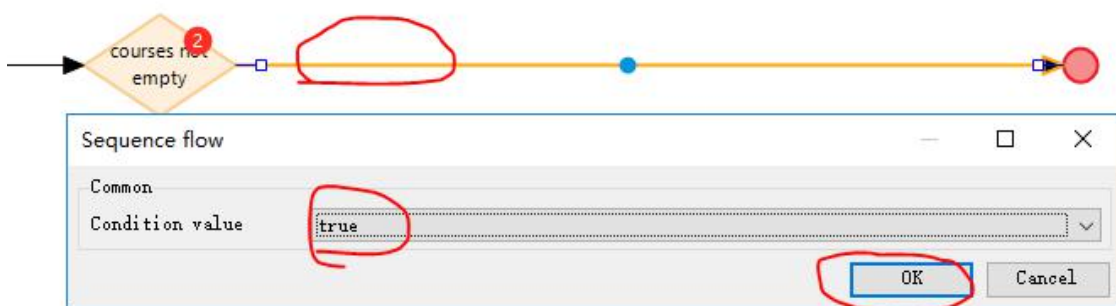
OK Cancel

- 我们在目录树中双击我们刚创建的微流 ACT_TrainingEvent_Save，然后拖入一个判断分支，然后双击打开，我们首先需要判断课程不为空，然后点击 Ok 按钮(不建议复制)。

`$TrainingEvent/MyFirstModule.TrainingEvent_Courses != empty`



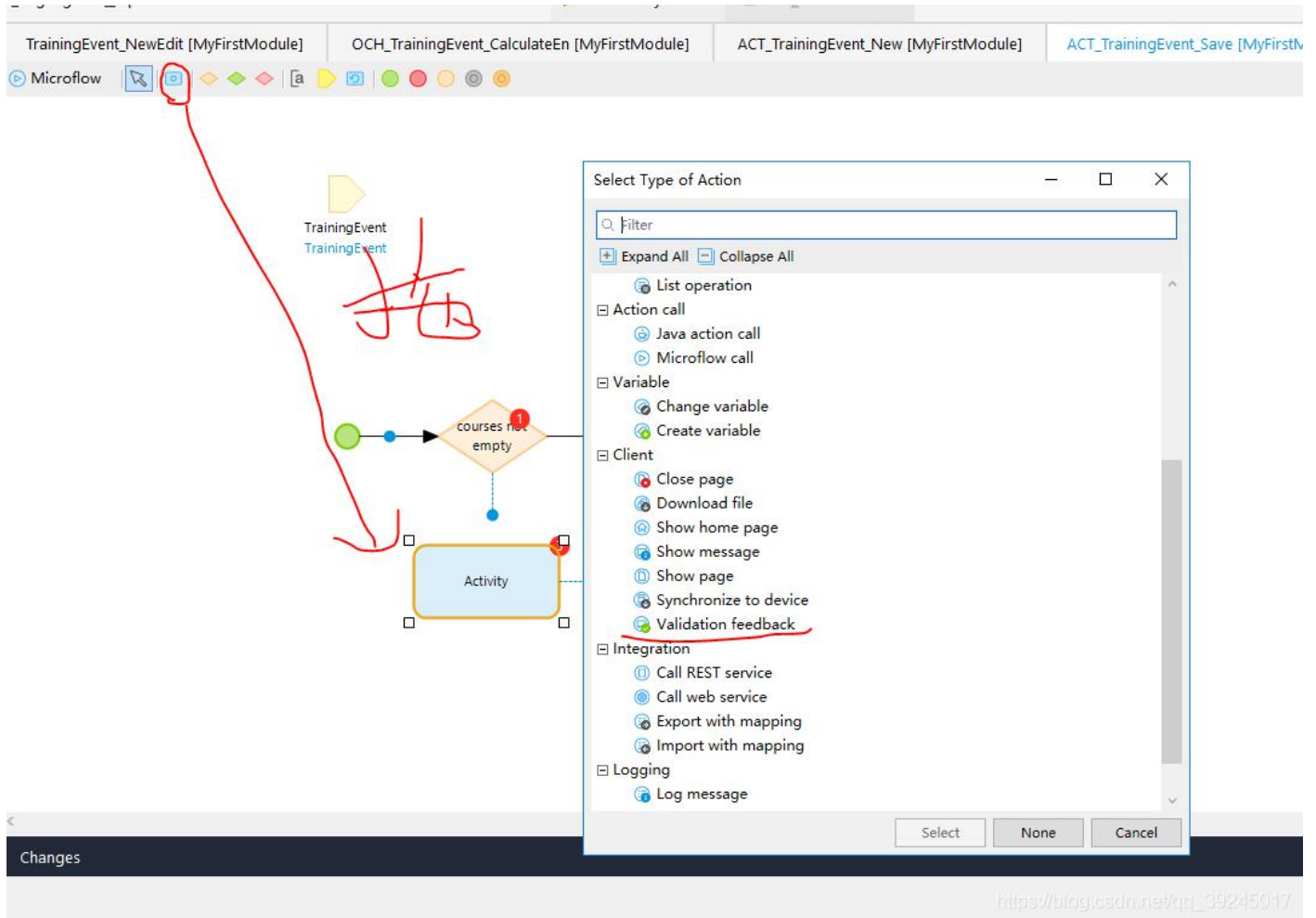
4. 判断分支设置好以后，我们需要给他设置流向，双击下图红色区域，选择 True，然后选择 Ok 按钮，不为空时我们让他继续。



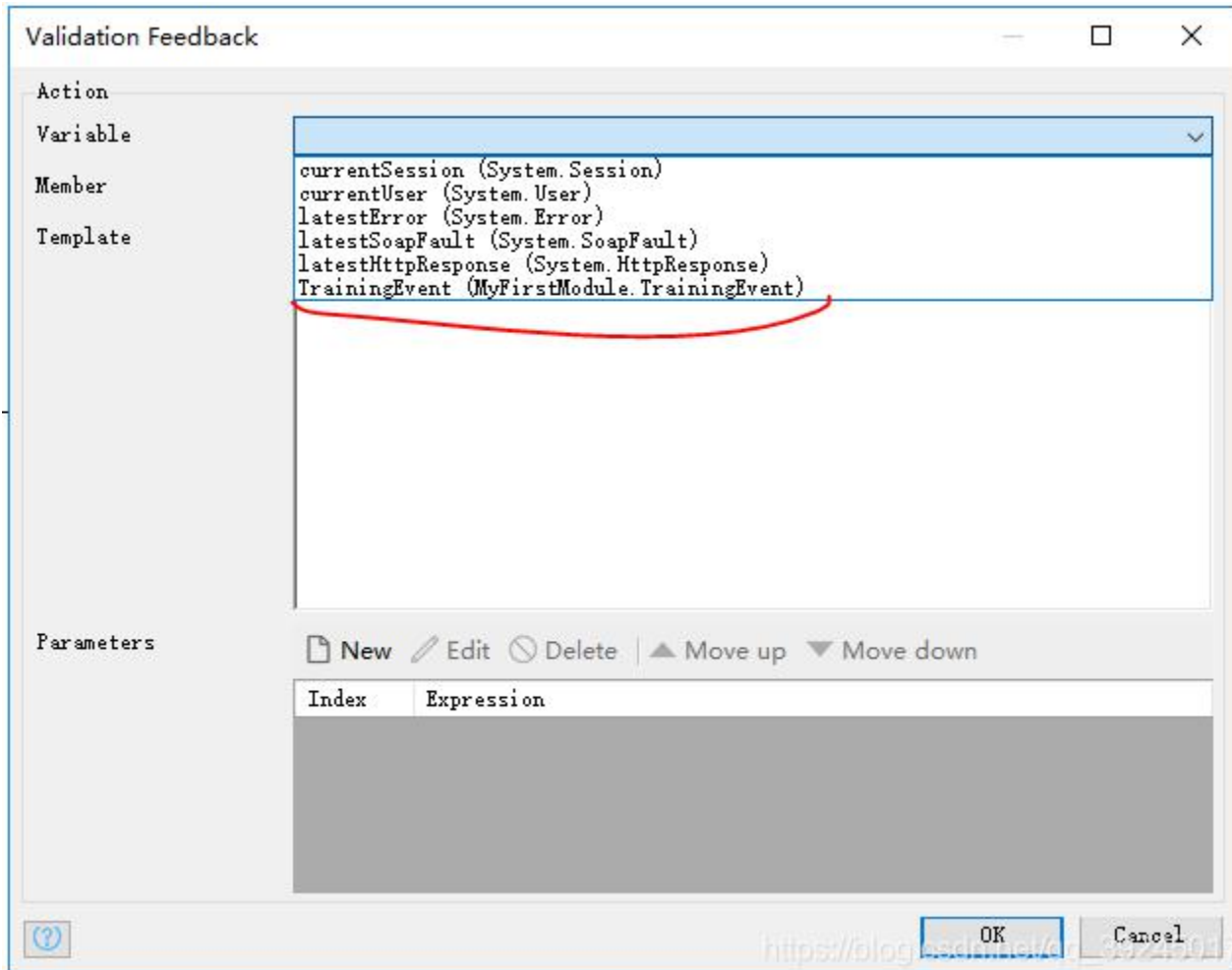
https://blog.csdn.net/qq_39245017

5. 我们需要在为空的时候在输入框显示验证的信息，所以需要再拖入一个 activity 事件，我们选择

Client(客户端)的 Validation feedback(验证模块)，来进行验证，选择好以后点击 Select 按钮。



6. 然后在弹出的页面中，我们选择 TrainingEvent(活动数据模型)。



7. 然后我们选择课程文本框。

Validation Feedback

Action

Variable

Member

Template

Parameters

TrainingEvent (MyFirstModule.TrainingEvent)

startDate

endDate

NrOfRegistrations

TrainingEvent_Courses

TrainingEvent_Teacher

TrainingEvent_Location

New

Edit

Delete

Move up

Move down

Index

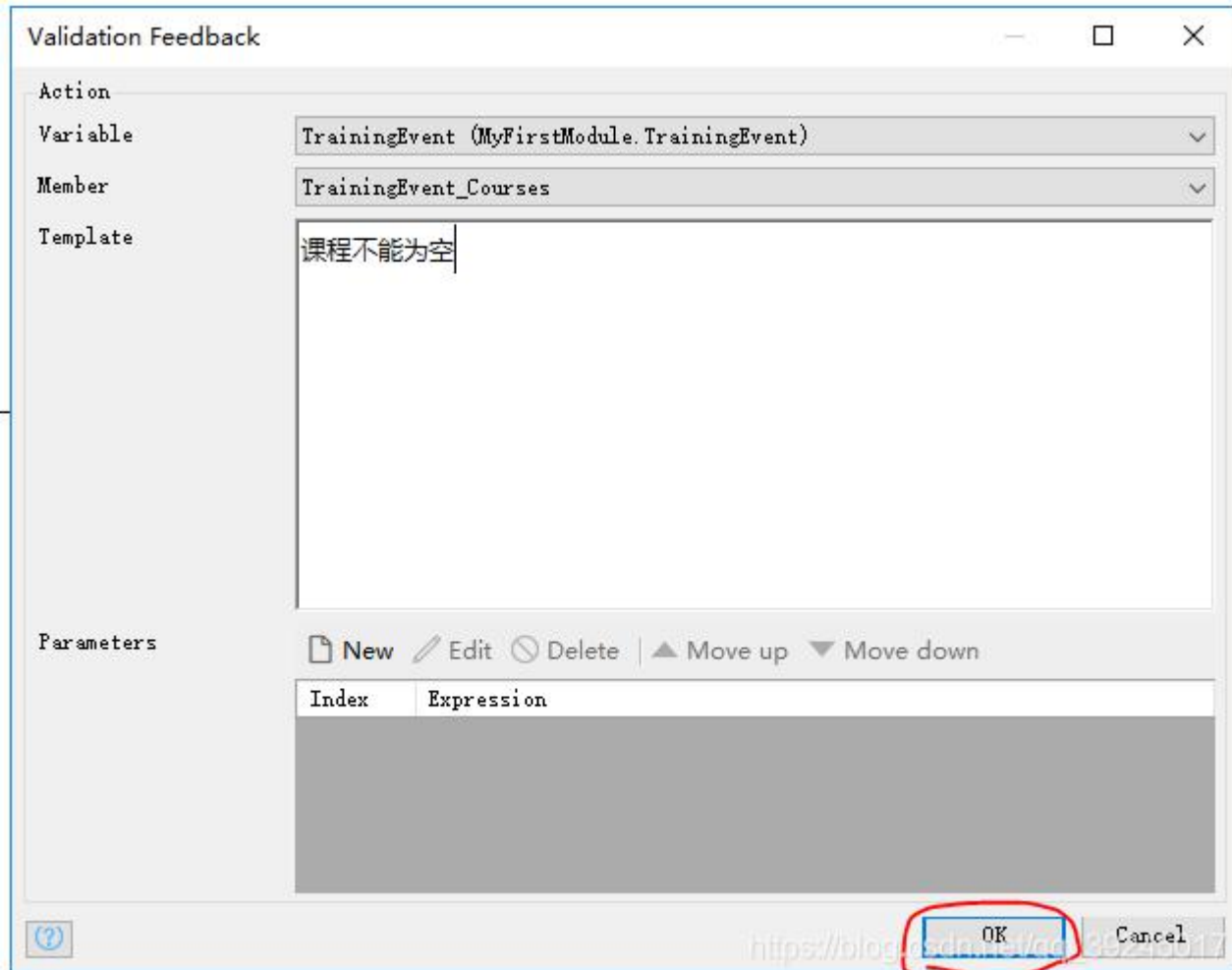
Expression

?

OK

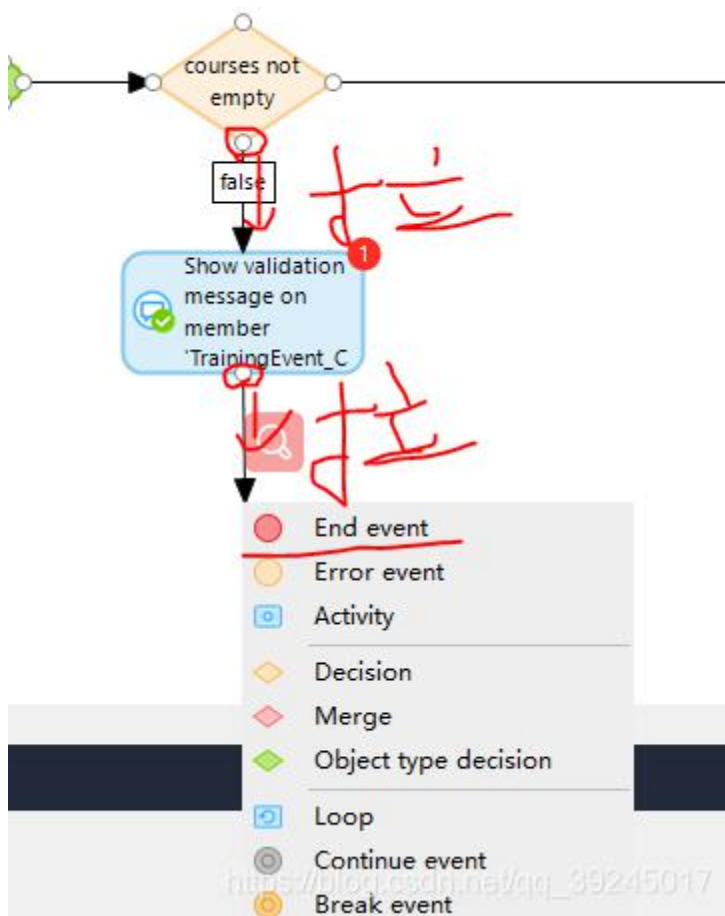
Cancel

8. 然后在 Template 中输入需要提示的信息，然后点击 Ok 按钮。



9.这个时候我们需要判断流向为 false 拉向我们刚创建的提示文本，然后提示完以后走向就结束了。

所以需要再拉一个走向结束的节点。鼠标左键拉，松开以后就会让你选择，我们选择 End event 节点



9. 到这一步，我们就把课程非空校验已经做完了，接下来我们需要做开始日期、老师、地址的非空判断，我就不一一罗列了。操作步骤是一样的，所以我只展示关键的信息。

10. 开始时间非空处理

11. `$TrainingEvent/startDate != empty`

Decision

Common

Caption

startDate != empty

Decision Type

☒ Expression
 ☐ Rule

Expression

Expression wizard

\$TrainingEvent/startDate != empty

Line	Column	Error

?

OK

Cancel

Validation Feedback

Action

Variable

Member

Template

Parameters

TrainingEvent (MyFirstModule.TrainingEvent)

startDate

开始时间不能为空

New

Edit

Delete

Move up

Move down

Index	Expression

?

OK

Cancel

11. 老师非空处理: `$TrainingEvent/MyFirstModule.TrainingEvent_Teacher != empty`

Decision

Common

Caption

Teacher != empty

Decision Type

☒ Expression
 ☐ Rule

Expression

Expression wizard

`$TrainingEvent/MyFirstModule.TrainingEvent_Teacher != empty`

Line	Column	Error

?

https://blog.csdn.net/qq_39249917

OK

Cancel

Validation Feedback

Action

Variable

Member

Template

Parameters

TrainingEvent (MyFirstModule.TrainingEvent)

TrainingEvent_Teacher

老师不能为空

New

Edit

Delete

Move up

Move down

Index	Expression

OK

Cancel

12. 地址非空处理 `$TrainingEvent/MyFirstModule.TrainingEvent_Location != empty`

Decision

Common

Caption
Location != empty

Decision Type
☒ Expression
☐ Rule

Expression

Expression wizard

\$TrainingEvent/MyFirstModule.TrainingEvent_Location != empty

Line	Column	Error

(?)

<https://blogs.sap.com/2017/09/17/>

OK

Cancel

Validation Feedback

Action

Variable

Member

Template

TrainingEvent (MyFirstModule.TrainingEvent)

TrainingEvent_Location

地址不能为空

Parameters

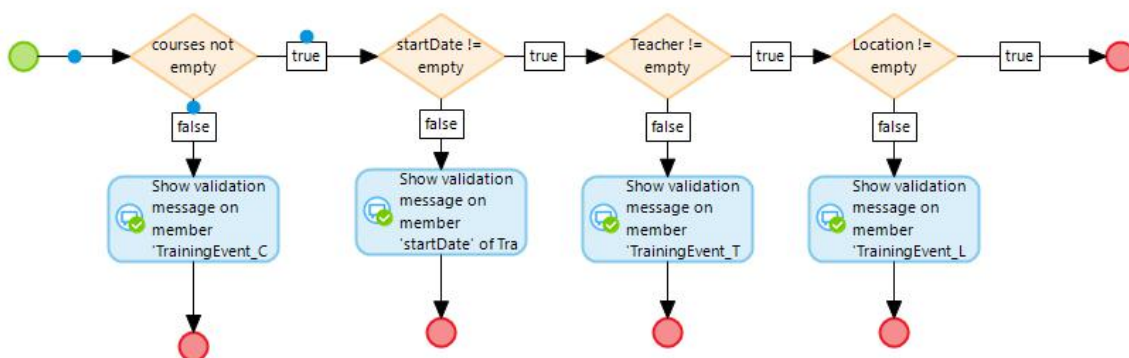
New Edit Delete Move up Move down

Index	Expression

OK Cancel

13. 整体非空处理完成以后的走向图展示。

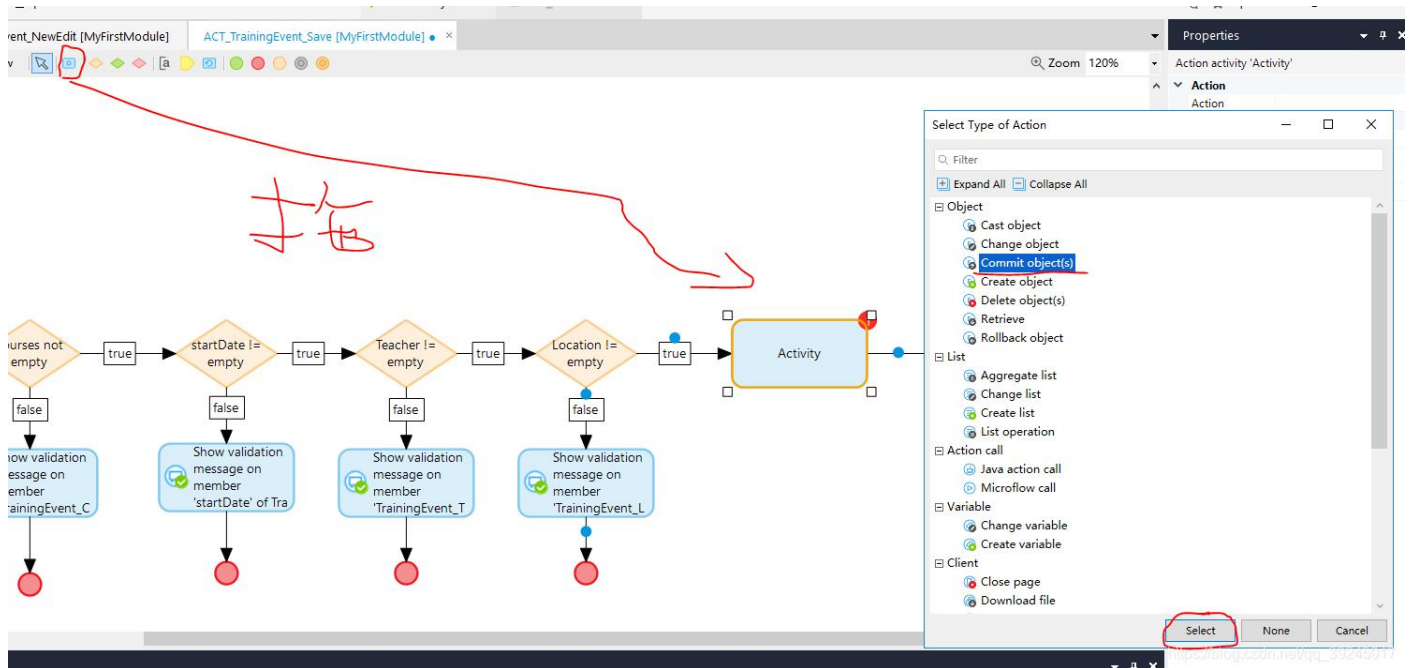
TrainingEvent
TrainingEvent



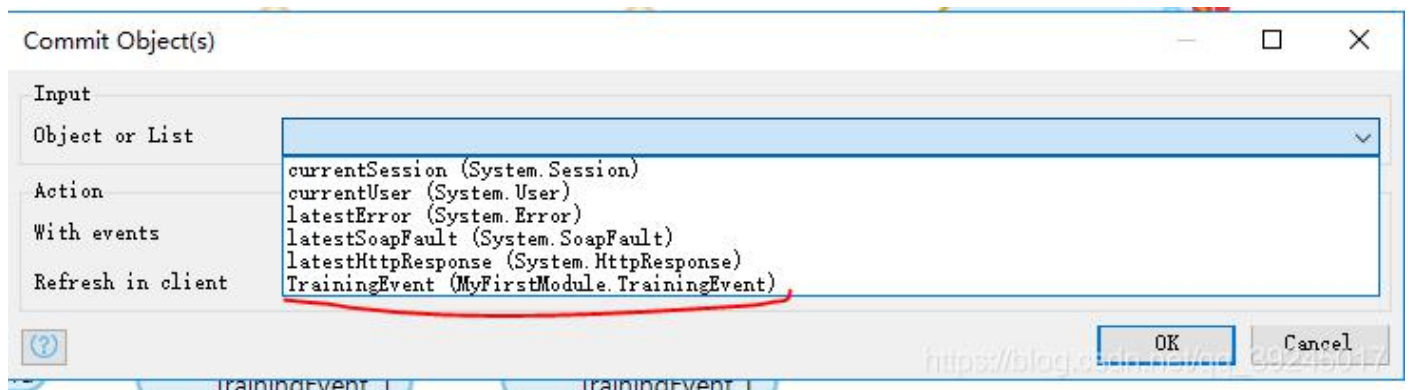
https://blog.csdn.net/qq_39245017

14. 非空判断以后，我们需要模拟表单的提交操作，所以需要拖入一个 Activity 事件，双击我们选择 Commit

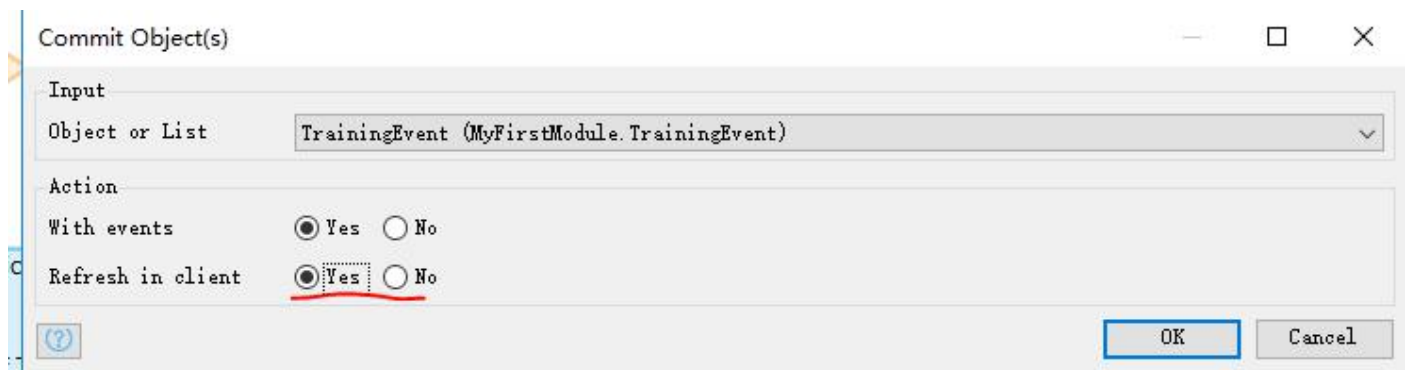
object(表单提交组件)，然后选择 Select 按钮



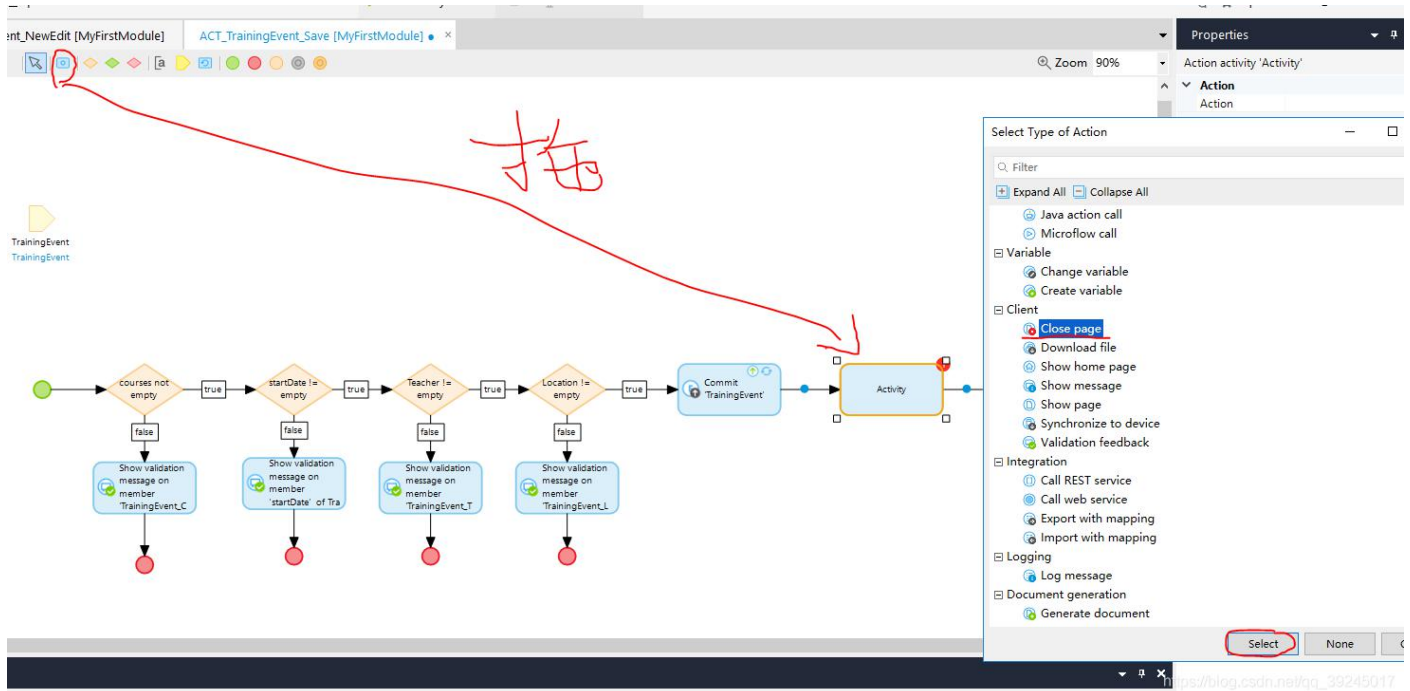
15. 我们在 Object or List 选择需要提交的模型为 TrainingEvent。



16. 我们勾选 Refresh in client 为 Yes(在客户端刷新)。然后点击 Ok 按钮。



17. 表单提交以后，我们模拟客户端关闭页面的动作，重新拖入一个 Activity 事件，双击选择 Close page 即可。



18. 我们运行项目，看一下效果。我们可以发现，他是一条一条的做验证，而不是我们所要的一下全部触发校验。

(文本框提示过高，是因为 Mendix 的一个 BUG，目前对中文有一些问题。我们可以在输入验证信息的文本框

中，将验证信息剪切出来，然后删除多余的空格以后保存，然后再次打开复制进去，就可以解决了)

Edit Training Event

Courses

课程不能为空

start date

m/d/yyyy

end date

m/d/yyyy

Teacher

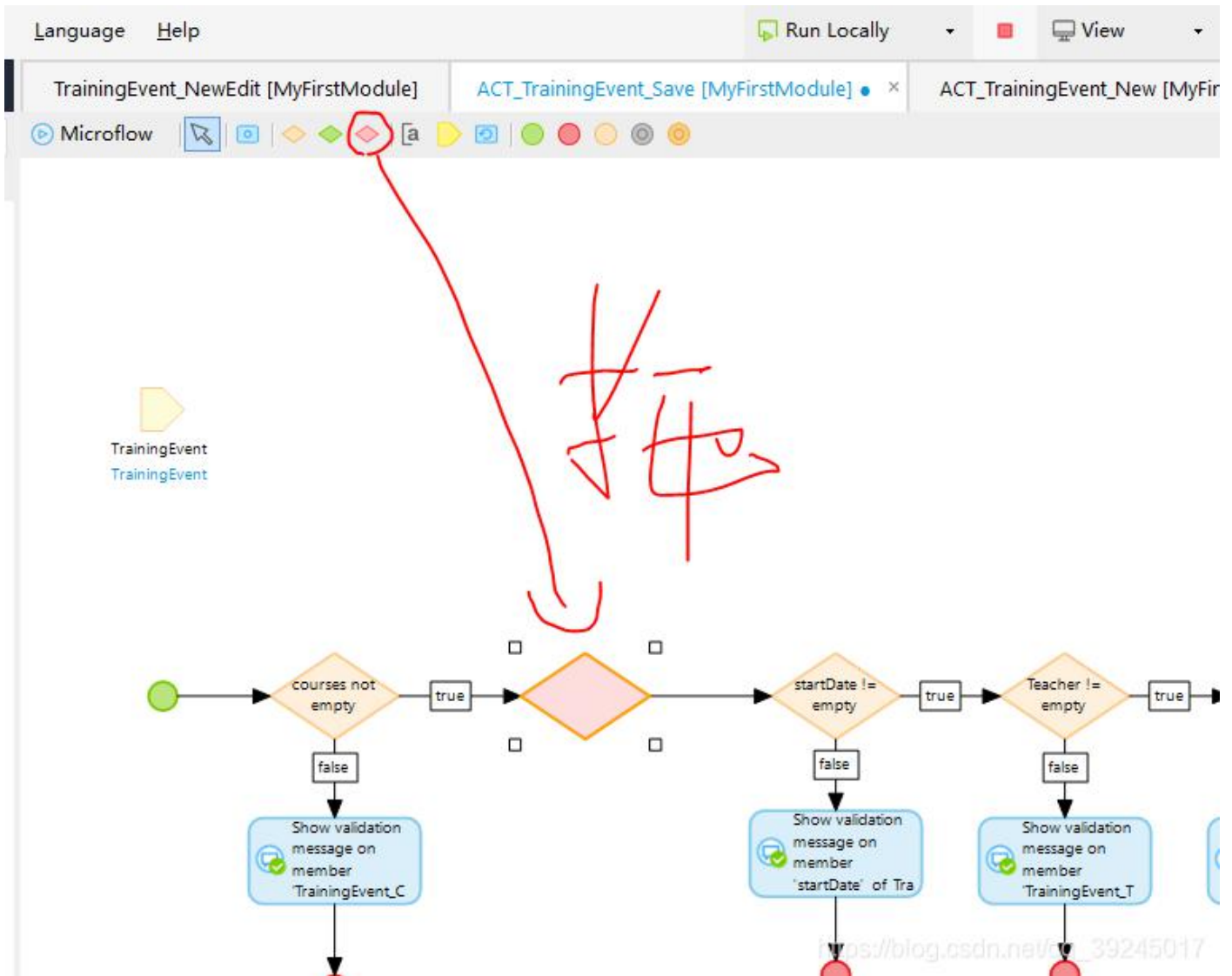
Location

Save

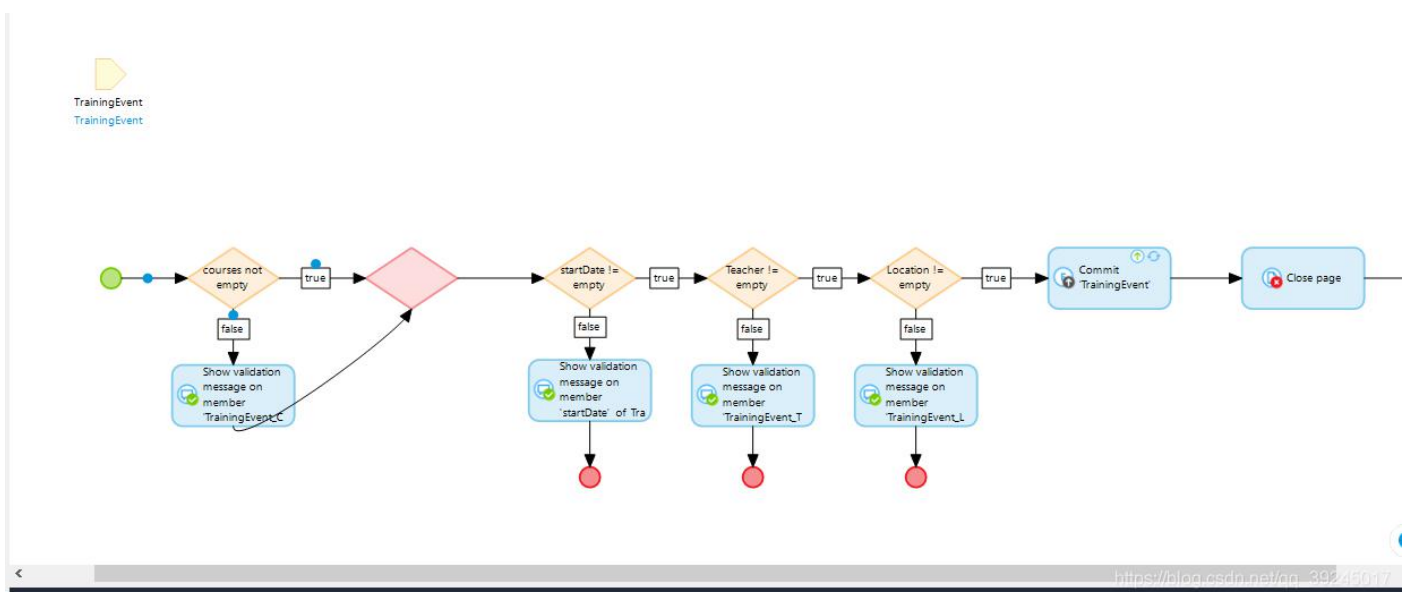
Cancel

https://blog.csdn.net/qq_39245017

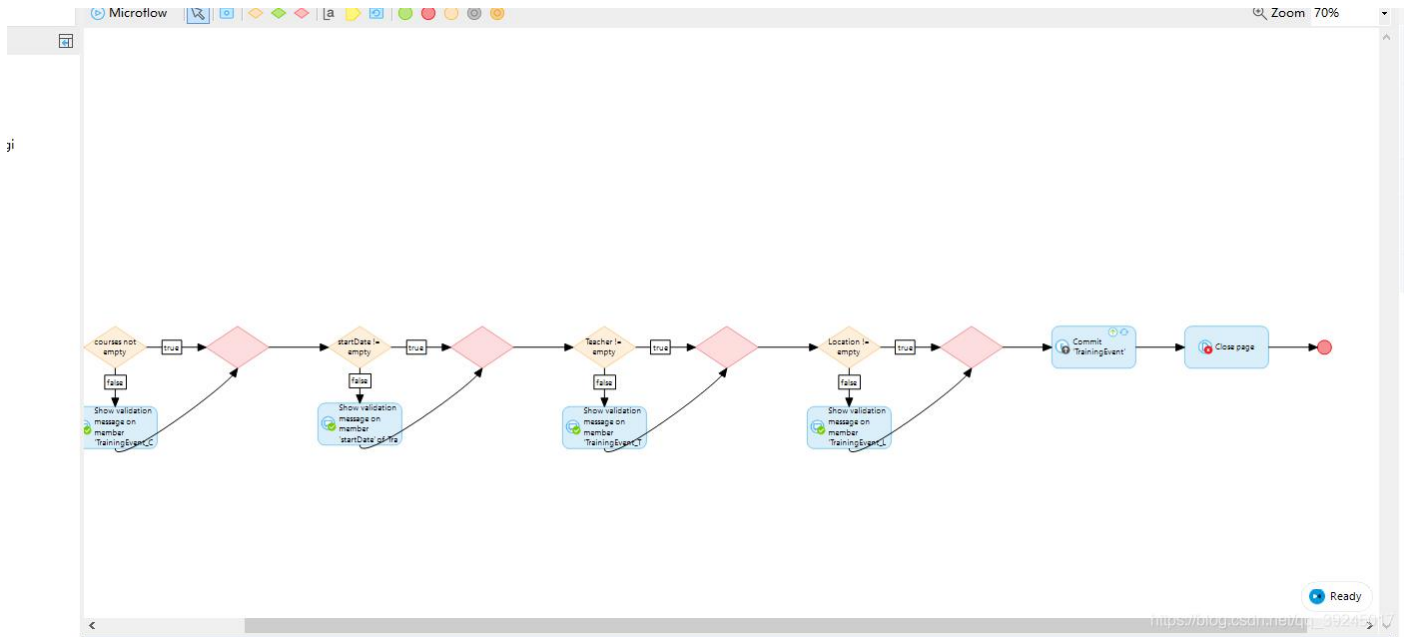
19. 我们现在处理一下，触发全部表单验证，首先拖入一个分支合并节点 Merge 到工作台。



20. 然后将我们的结束节点删除，将验证失败以后的流向，指到刚才拖入的分支里。



21. 依次将其余的验证失败分支这么处理。

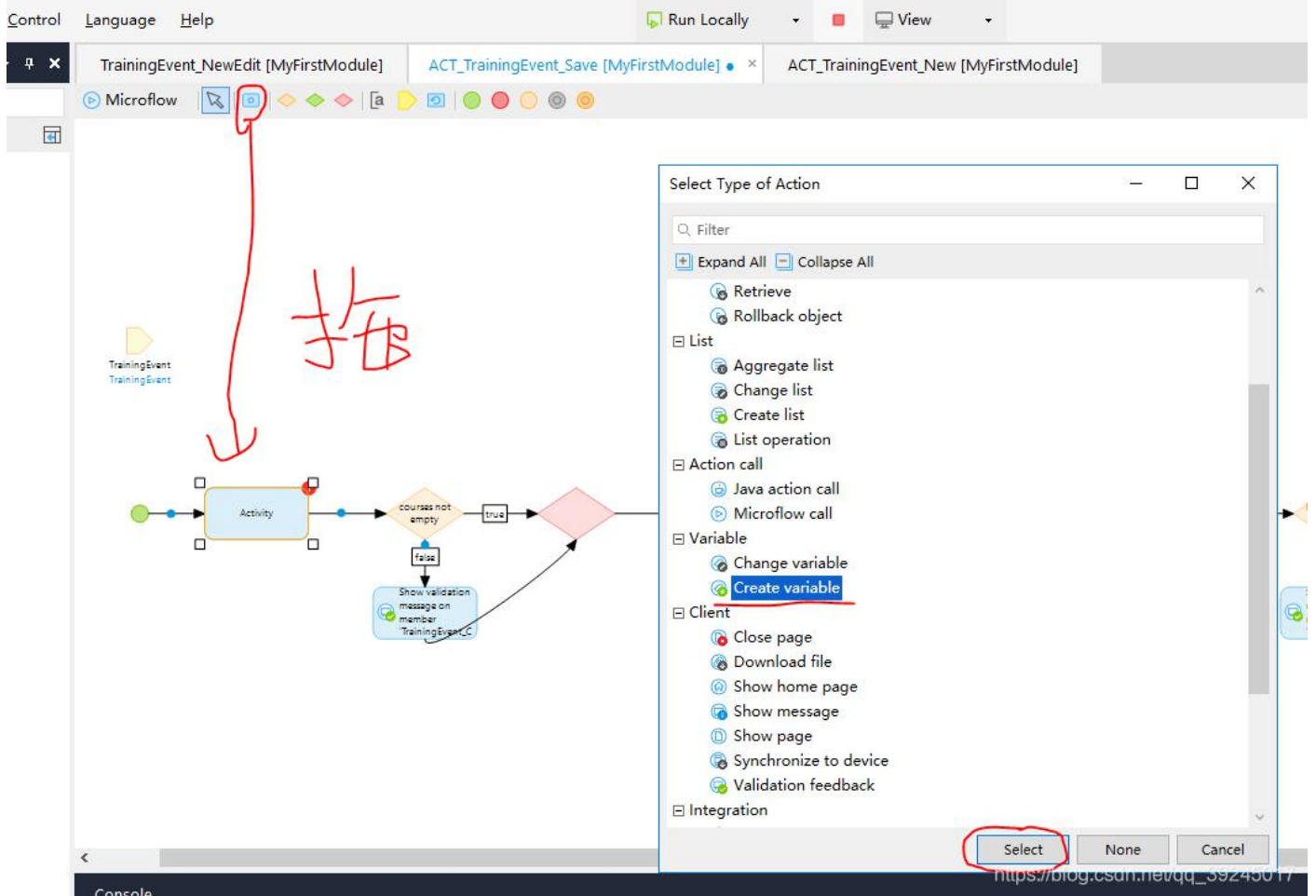


22. 然后我们保存，运行一下项目，再次验证一下效果。

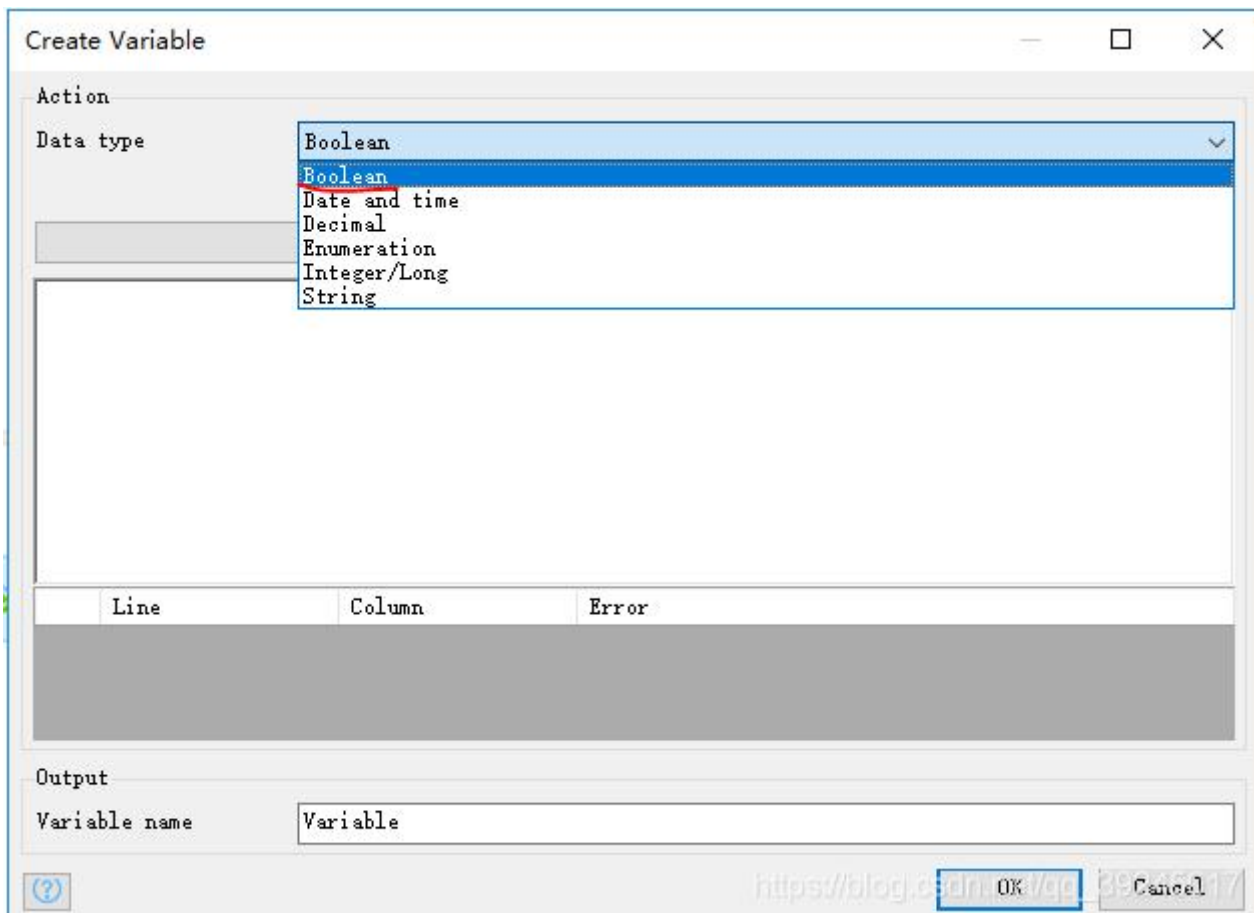
23. 我们可以发现，触发全部字段的验证并展示了错误信息以后程序接着又保存数据了。所以逻辑是不对的。

24. 我们可以在程序验证前加一个标记，每次验证的时候触发这个标记，最后提交的时候判断该标记是否可以提交。

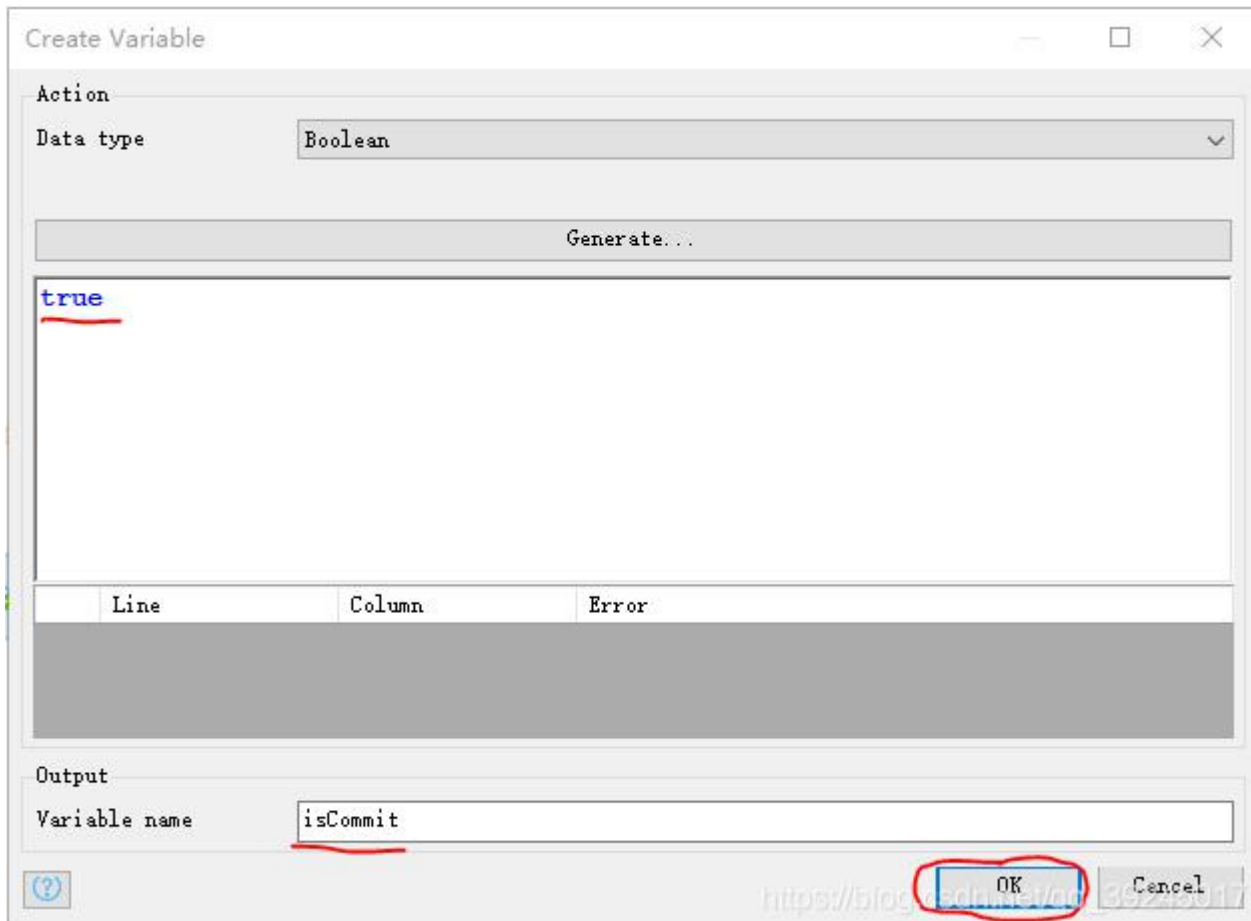
思路滤清，我们开始实现。首先需要拖入一个 Activity 时间，我们选择 Ccreate variable(创建一个临时变量)，然后点击 Select 按钮。



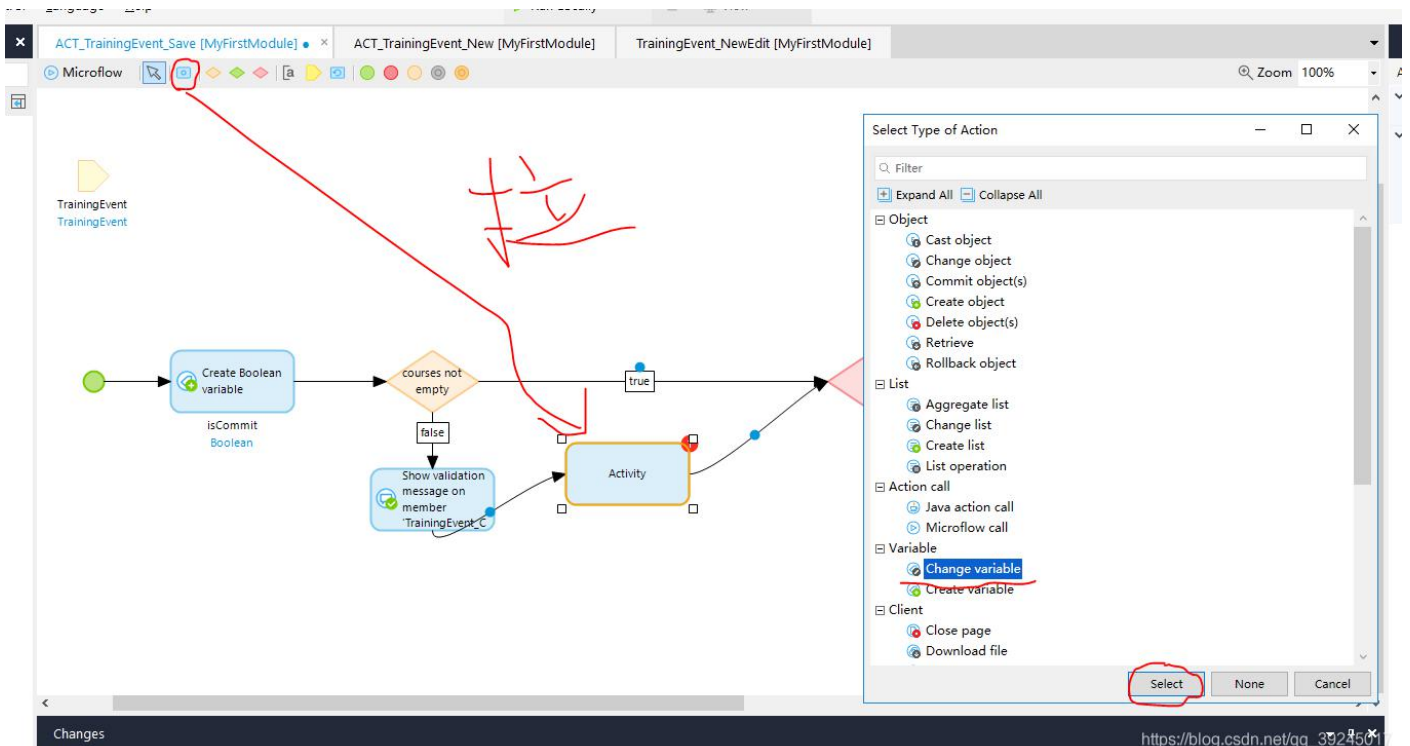
25. 因为我们只需要做判断，所以我们选择 Boolean 类型。



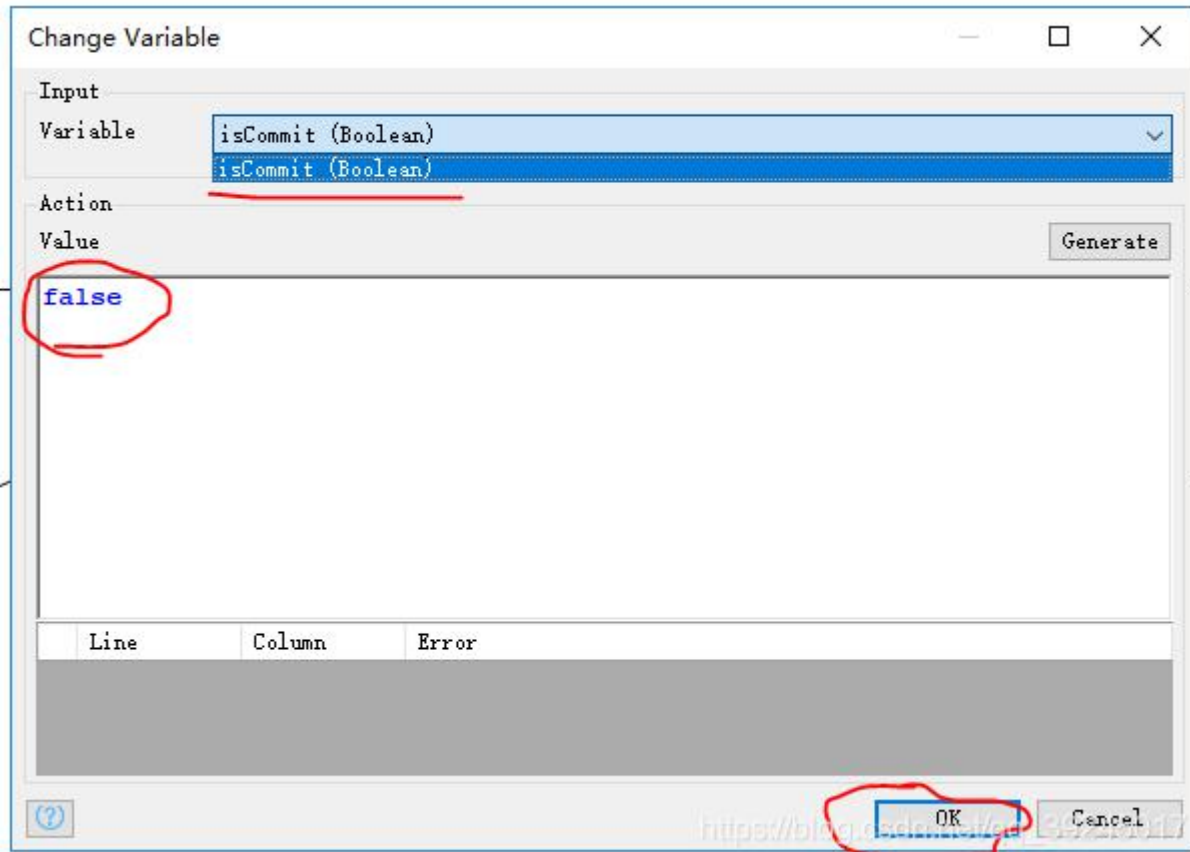
26. 默认设置 true 可以提交, 然后我们修改 Variable name 变量名称为 isCommit(能否提交), 然后点击 Ok 按钮。



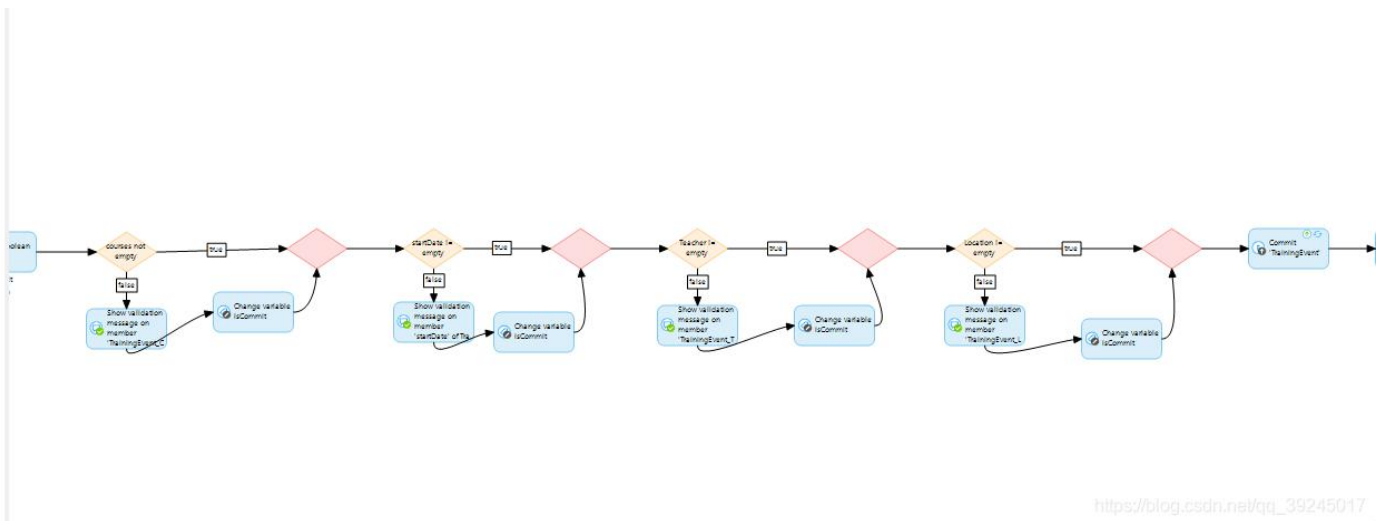
27. 设置完变量以后, 我们要在非空判断为空的情况追加 Activity, 然后改变我们的变量。拉入一个 Activity 事件, 然后选择 Change variable(改变临时变量), 然后点击 Select 按钮。



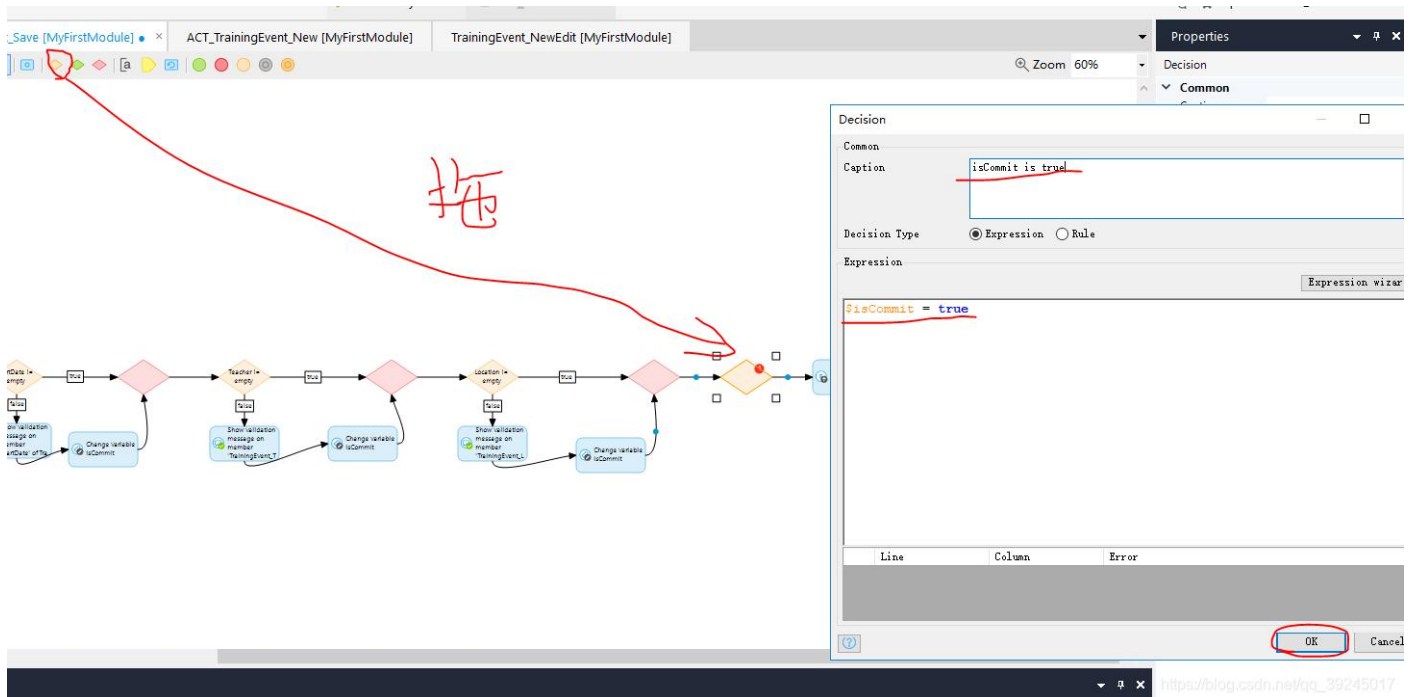
28. 然后将 isCommit 修改为 false，然后点击 Ok 按钮。



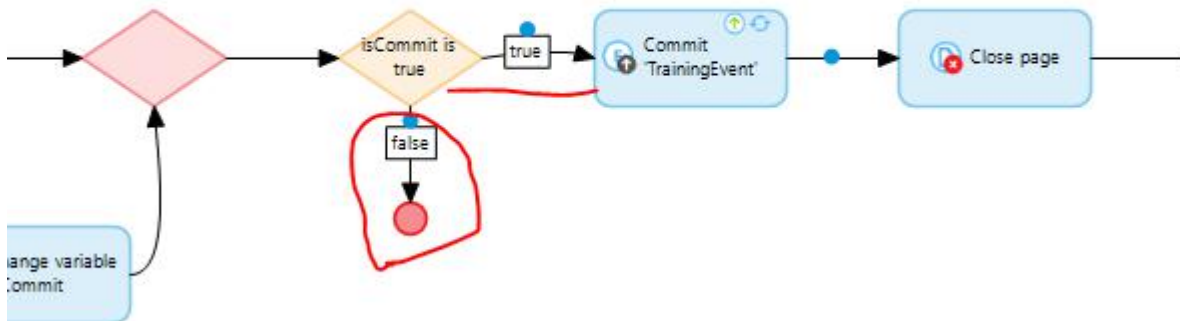
29. 同样的将其余的分支将 isCommit 修改为 false。修改后的效果。



30. 然后我们在最后的判断节点上追加条件分支，来进行判断 isCommit 是否可以提交。



31. 如果可以提交的话就继续执行，如果不可以提交的话就结束。



https://blog.csdn.net/qq_39245017

32. 保存项目然后运行，我们看效果。发现该效果已经实现了。

Edit Training Event

Courses

课程不能为空

start date

m/d/yyyy

开始时间不能为空

end date

m/d/yyyy

Teacher

老师不能为空

Location

地址不能为空

Save

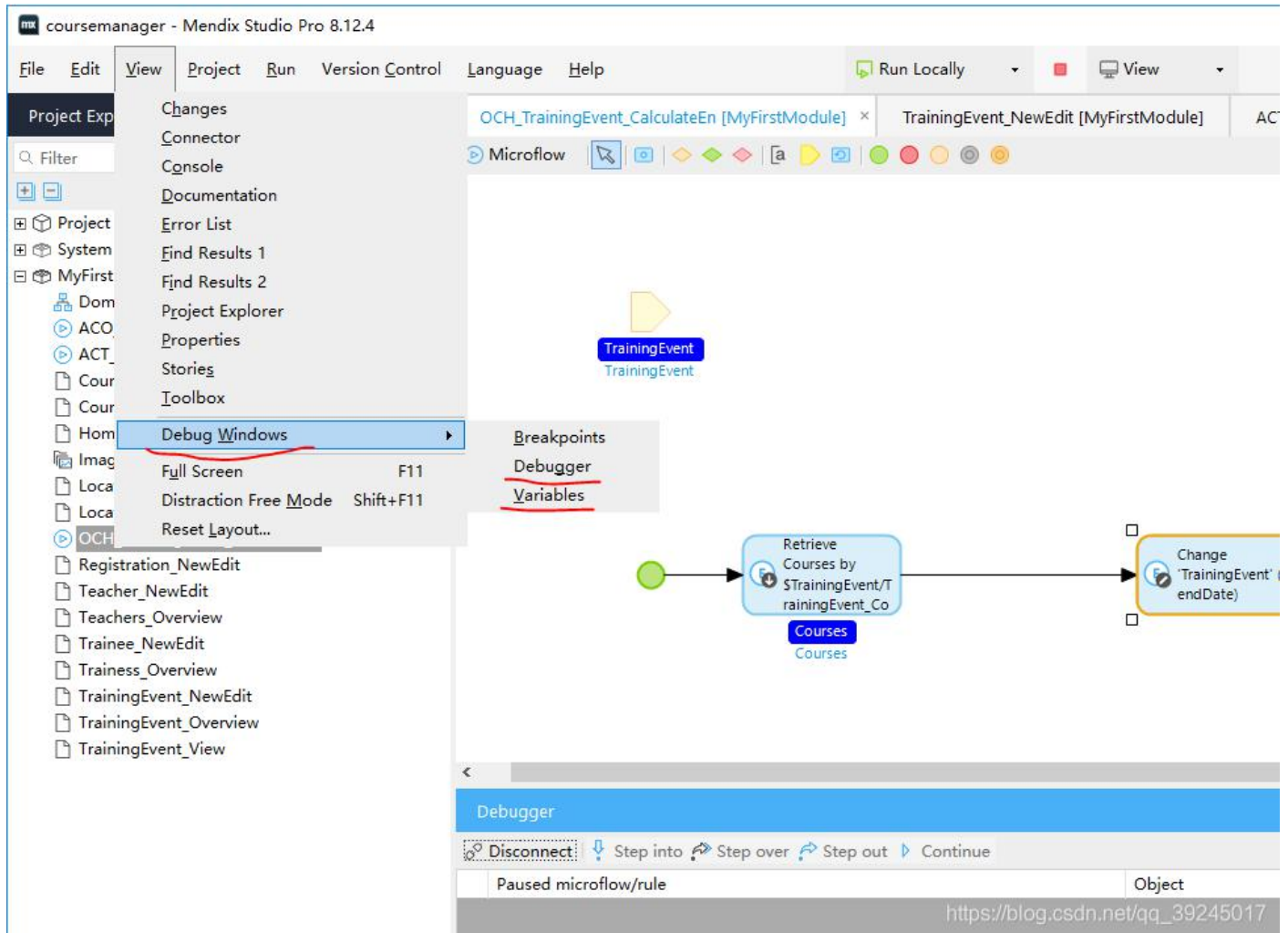
Cancel

https://blog.csdn.net/qg_39245017

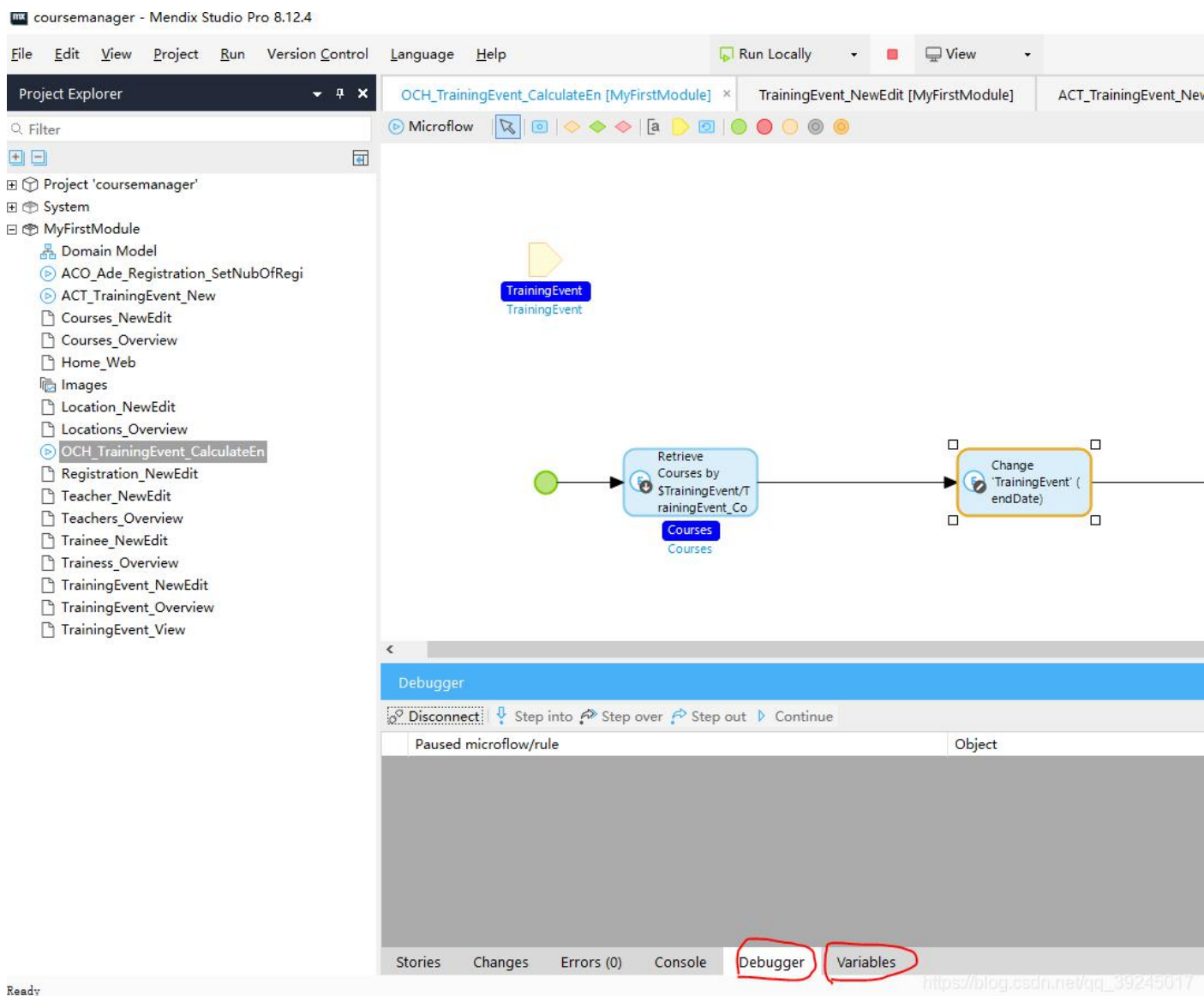
二、DeBug 问题跟踪

1、工具介绍

1. 首先我们将 Debugger 菜单调整出来，点击菜单栏 View，找到 Debug Windows(问题跟踪窗口)，然后点击 Debugger(跟踪栏)和 Variables(参数)



2. 这时候我们就可以在控制台中查看到 Debugger(跟踪栏)和 Variables(参数)两个选项栏。当您调试微流时，“Debugger”窗格会很有用，因为它将引导您完成微流的操作。

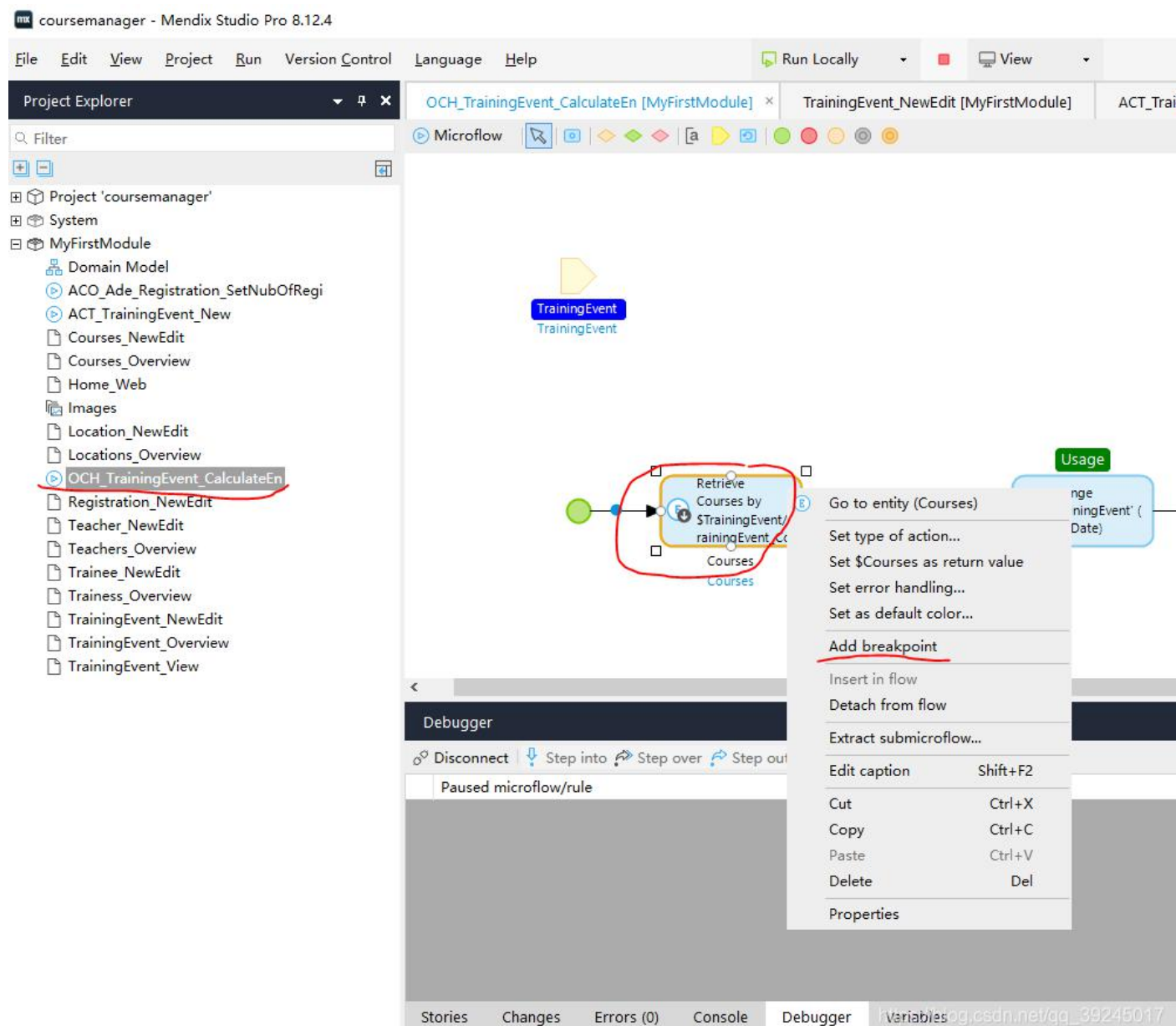


3. “Variables”窗格显示了微流中涉及的变量，对象和列表，以及当您通过“调试器”窗格逐步通过微流时它们如何更改。

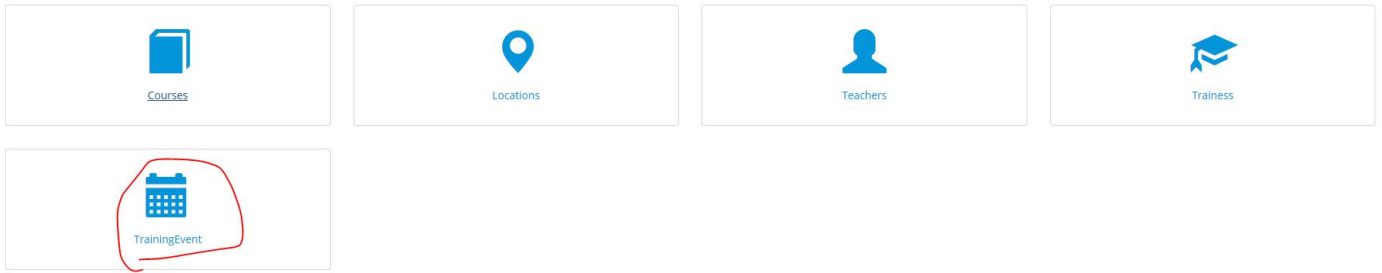
Variables		
Name	Type	Value
currentDeviceType	Enumeration 'System.Device...	Desktop
currentSession	System.Session	(id: 6473924464345289, state: normal)
id	Long	6473924464345289
createdDate	Date and Time	UTC time: 2018-10-15 12:50:16.258 Session time: 2018-10-...
CSRFToken	String	'e83f8138-0e2d-4351-9186-69597782cab1'
LastActive	Date and Time	UTC time: 2018-10-15 12:50:18.994 Session time: 2018-10-...
ReadOnlyHashKey	String	'a6a2aa0a-fdcc-4bf1-8e57-ae0a5d682f9d'
SessionId	String	'e8703290-c1b3-408b-9b14-126447ccd563'
System.Session_...	Reference	System.User (ID: 281474976710657)
currentUser	System.User	(id: 281474976710657, state: normal)
id	Long	281474976710657
Active	Boolean	true
Blocked	Boolean	false

2、问题查找

1. 经过对工具简单介绍以后，我们开始问题查找，首先需要在我们微流中打上一个断点，双击目录树上的 OCH_TrainingEvent_CalculateEn 微流组件，然后在第一步鼠标右击 上点击 Add breakpoint，增加断点。



2. 然后我们运行项目，运行我们刚才的程序。



https://blog.csdn.net/qg_39245017



JAVA高级教程
邵群超
1/14/2021
报名人数:0

Details 新增 报名

Edit Training Event

Courses

JAVA高级教程

start date

m/d/yyyy

end date

m/d/yyyy

Teacher

Location

Save

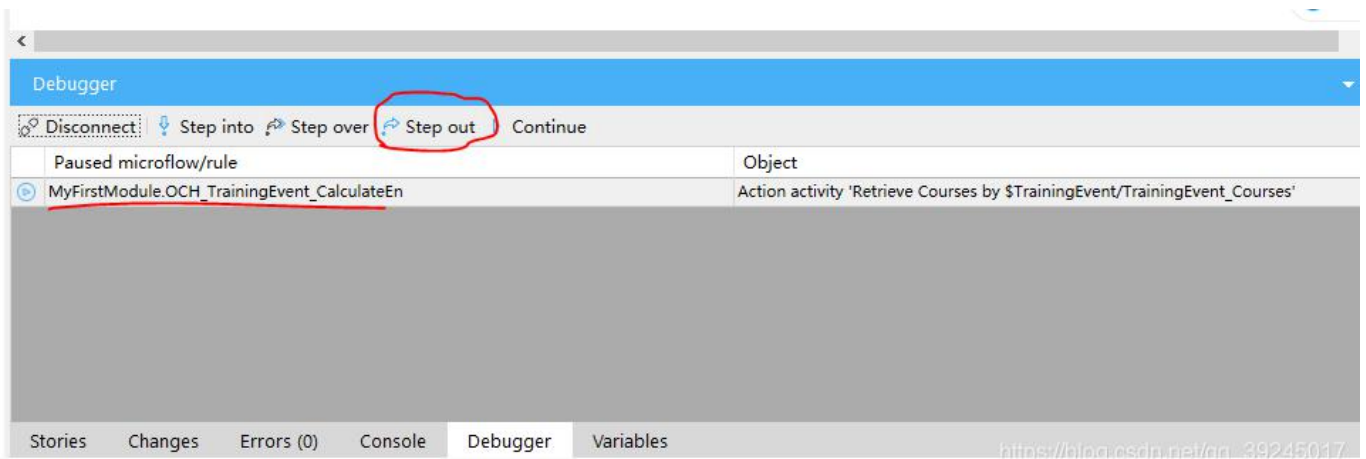
Cancel

https://blog.csdn.net/qg_39245017

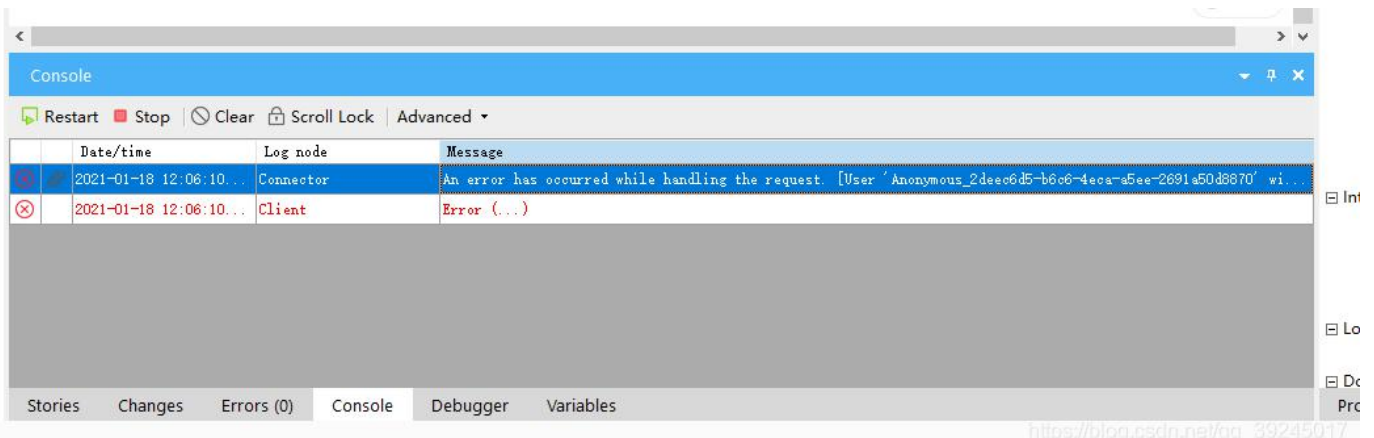
- 然后在桌面的导航栏中 Mendix 开发工具会有闪烁，证明我们刚才打的 Debugger 起作用了。然后我们打开我们的开发工具。



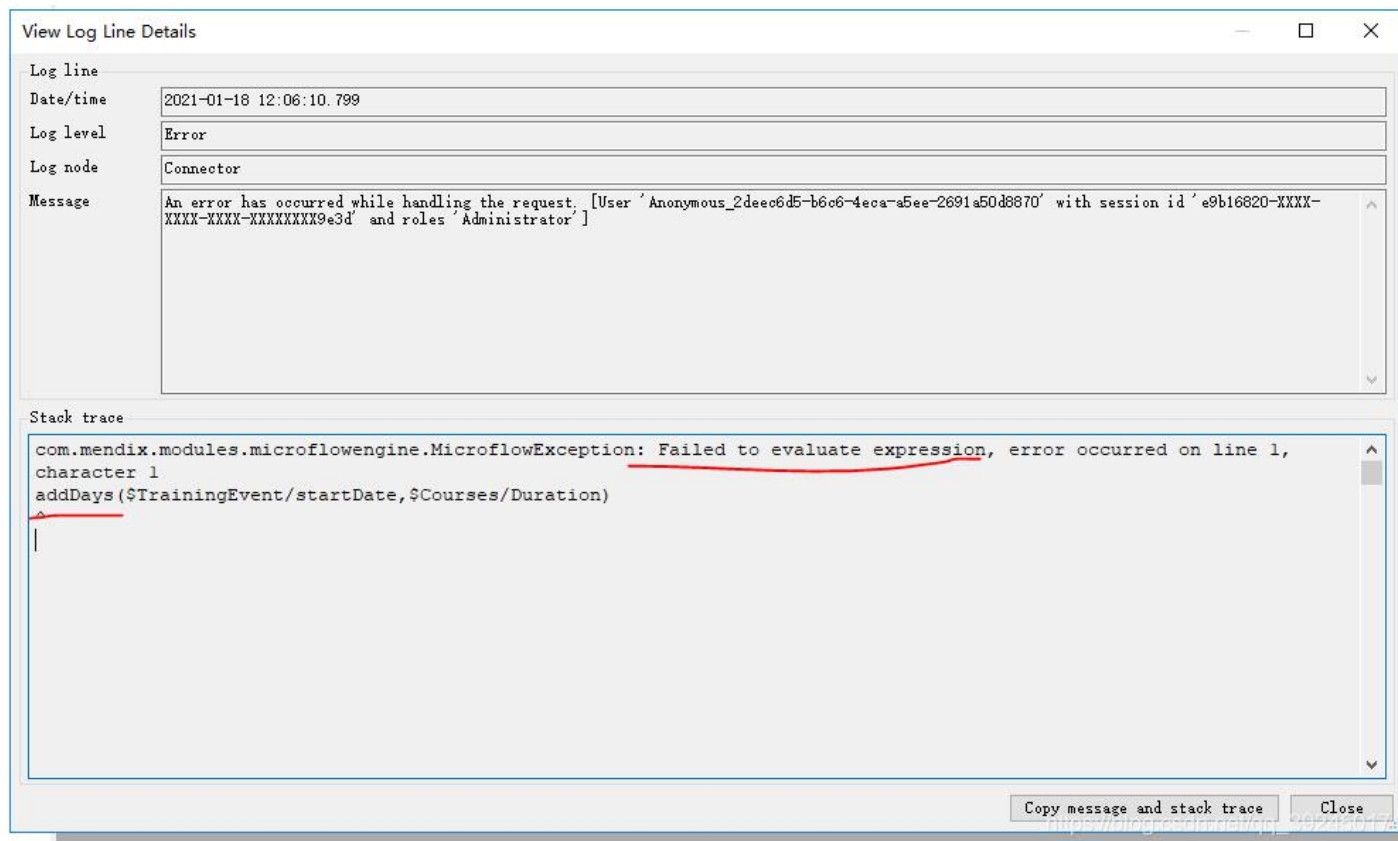
4. 这个时候中我们可以看到 Debugger 跟踪框中，会有一个跟踪的问题。我们点击 Step out 先让他运行结束，我们看控制台 Console 报什么错误。



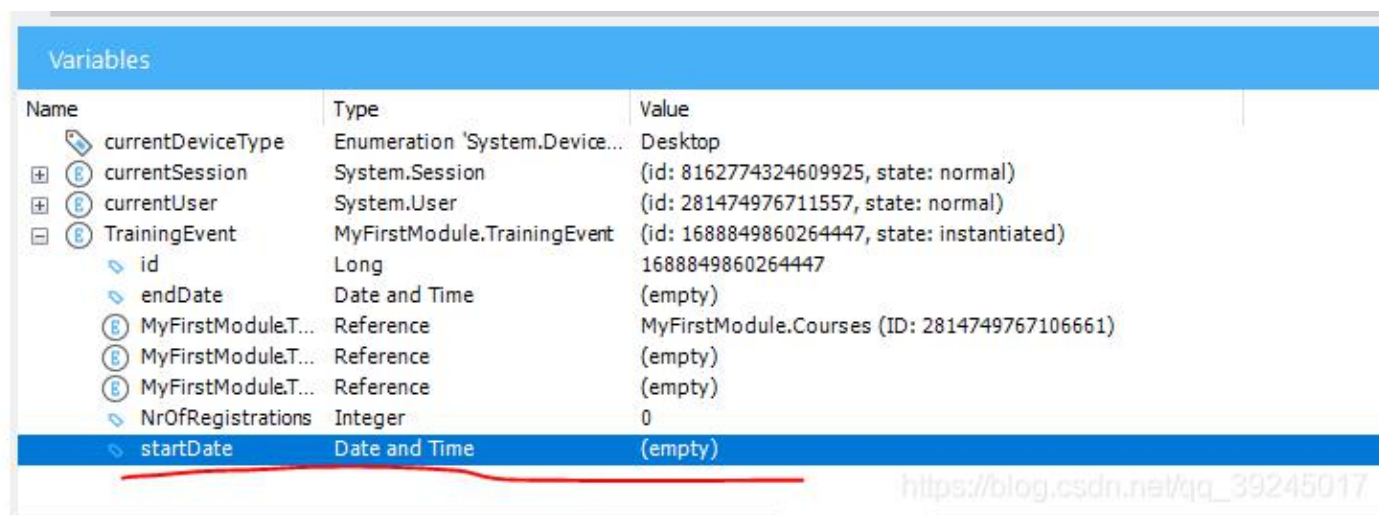
5. Connector 是程序后端的错误，Client 是前端的报错信息。我们双击点开 Connector 的错误，看看报了什么错。



6. 可以看到 addDays 运算出错，但是我们这个是运算成功的啊。那么我们分析一下，是不是计算空数据报的错误。



7. 然后我们再运行一遍，点开“Variables”窗口，可以看到 TrainingEvent(活动)数据模型中的 startDate(开始时间)是 empty 空的数据。



8. 经过问题跟踪，我们得知，addDays 在计算空的数据时会报错。所以我们需要在计算前做一下非空的处理。问题跟踪以后，我们继续来解决问题：[一、微流【3.2.2 结束日期统计\(14~21\)】](https://blog.csdn.net/qj_39245017)

三、数据嵌套

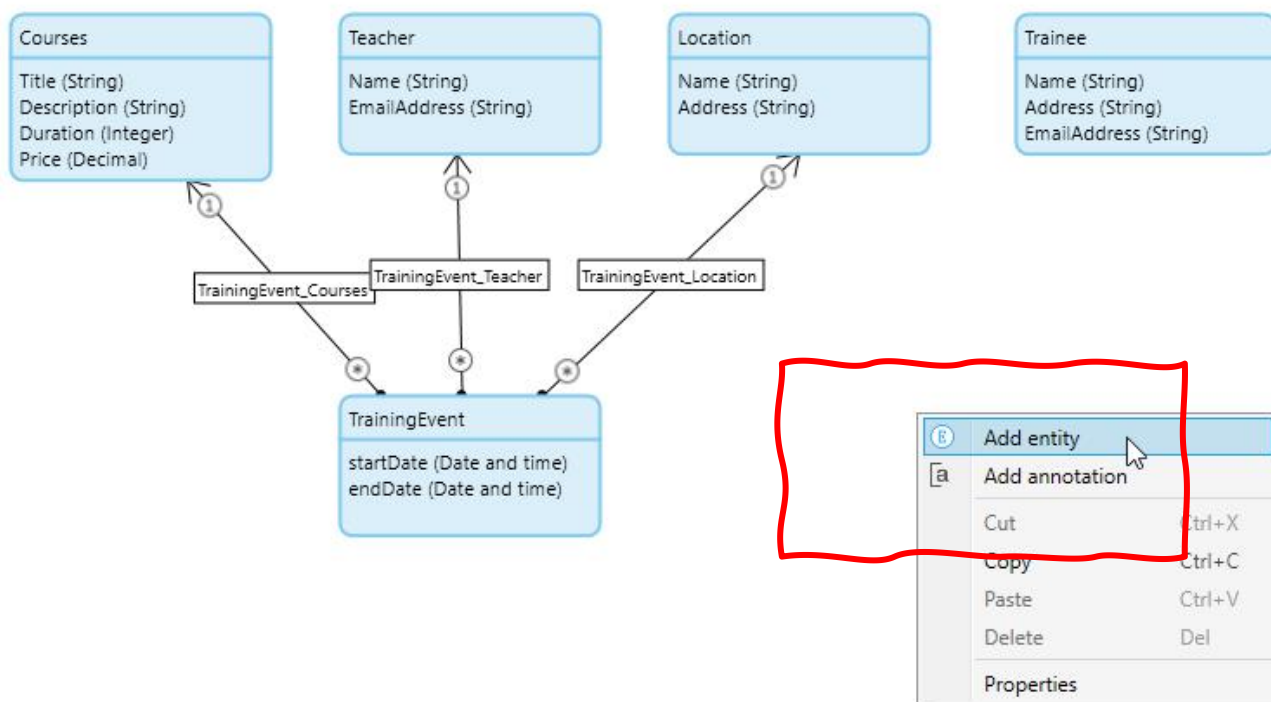
1、需求分析

学校需要在活动列表中有一个学生报名的页面。根据需求设计页面，上方需要展现活动详情，旁边有一个新增按钮，可以新增报名人员。下方是一个列表页，展示已经录入的报名信息。

2、项目实施

2.1 创建数据模型

1. 首先，需要创建一个报名人员数据模型。双击目录树中的 **Domain Mode**，右键单击画布空白处，选择 **Add Entity**，双击打开。



2. 在新弹出的页面我们修改 Entity **Name** 为 **Registration**，然后点击 **New** 增加属性 Attributes。

Properties of Entity 'MyFirstModule.Entity'

General

Name: Registration

Generalization: (none) [Select.]

Image: (none) [Select.]

Persistable: ☒ Yes ☐ No

Objects of this entity can only be stored in the database if it is persistable.

System members

☐ Store 'createdDate'

☐ Store 'changedDate'

☐ Store 'owner'

☐ Store 'changedBy'

Documentation

Attributes Associations Validation rules Event handlers Indexes

New [Insert new above selected] [Edit] [Delete] [Move up] [Move down]

Name	Type	Default value	Microflow
------	------	---------------	-----------

OK Cancel

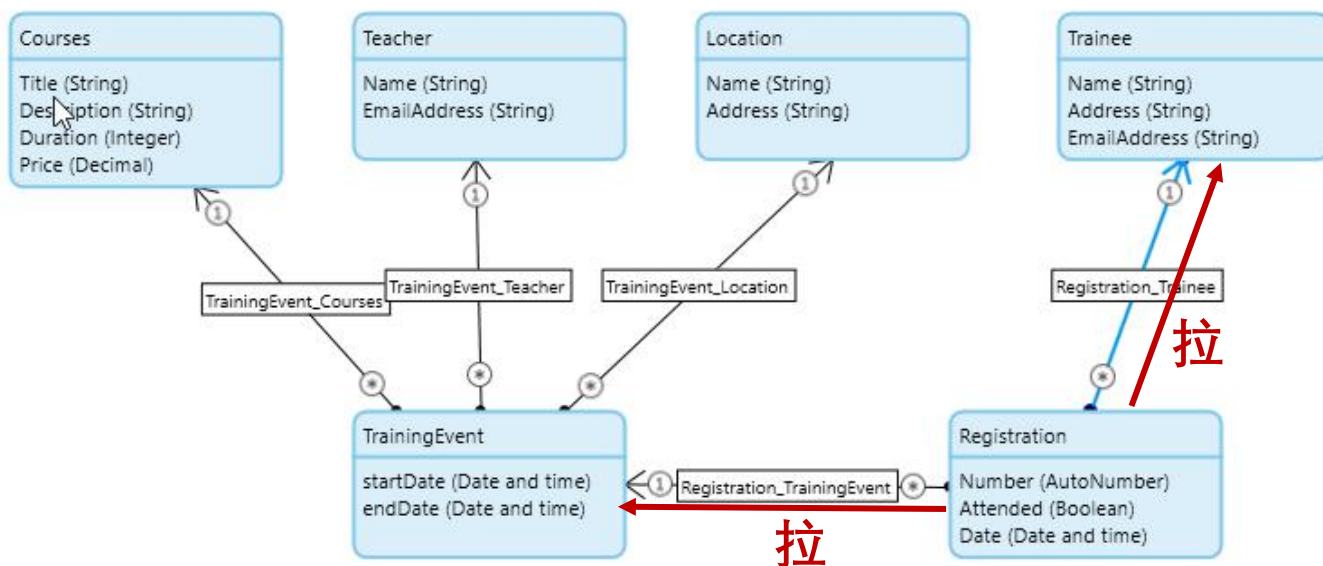
3. 以下是需要增加的属性，忘记的可以回顾【[Mendix 敏捷开发零基础学习《一》](#)】。

报名人员 Registration:

属性	备注	类型
RegisterID	ID	AutoNumber
Attended	是否报名	Boolean (<u>Default value</u> 选择 False，默认未报名)
RegDate	报名时间	Date and time(<u>Default value</u> 选择[%CurrentDateTime%]，系统默认时间)

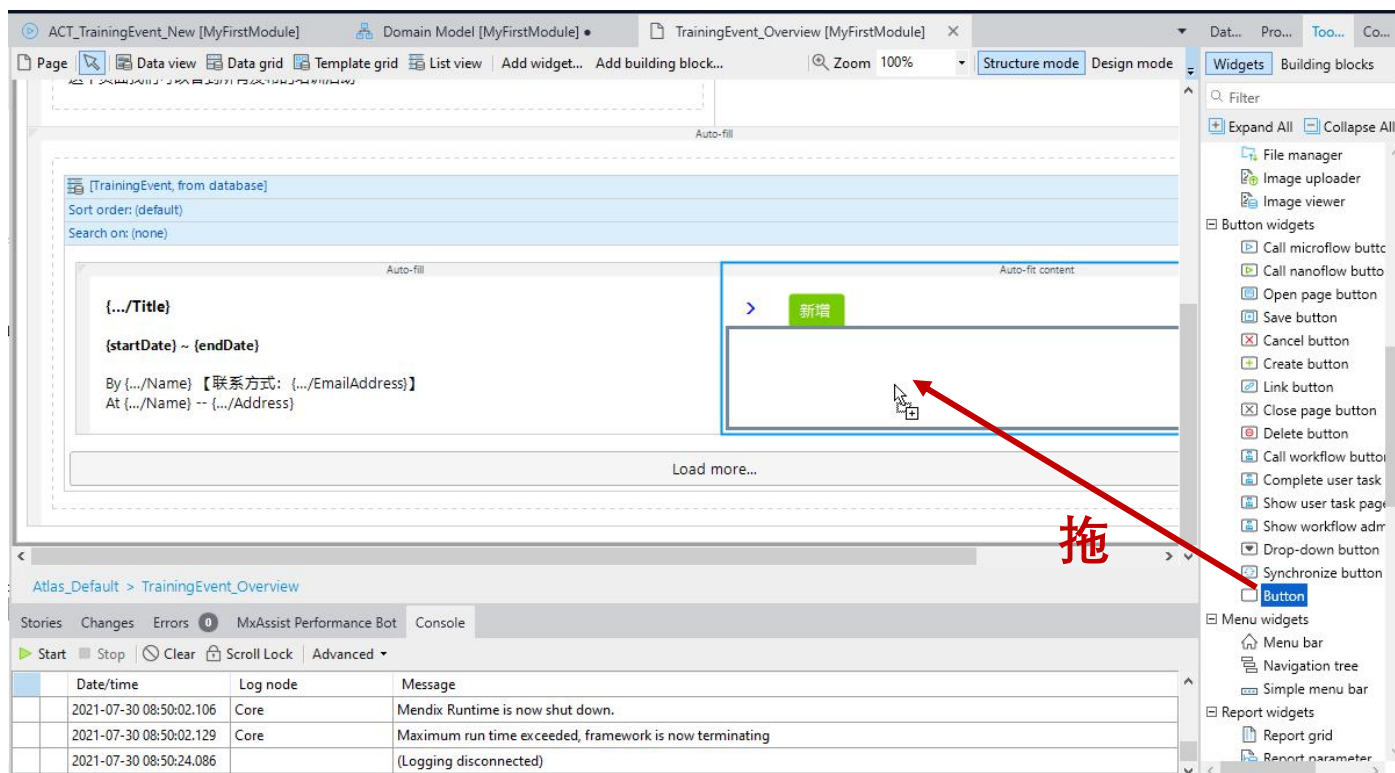
2.2 配置模型关系

- 接着，将报名人员 **Registration** 数据模型与 **Trainee(学生)**和 **TrainingEvent(活动)**进行关联。如图所示，直接拉两条线就可以了。

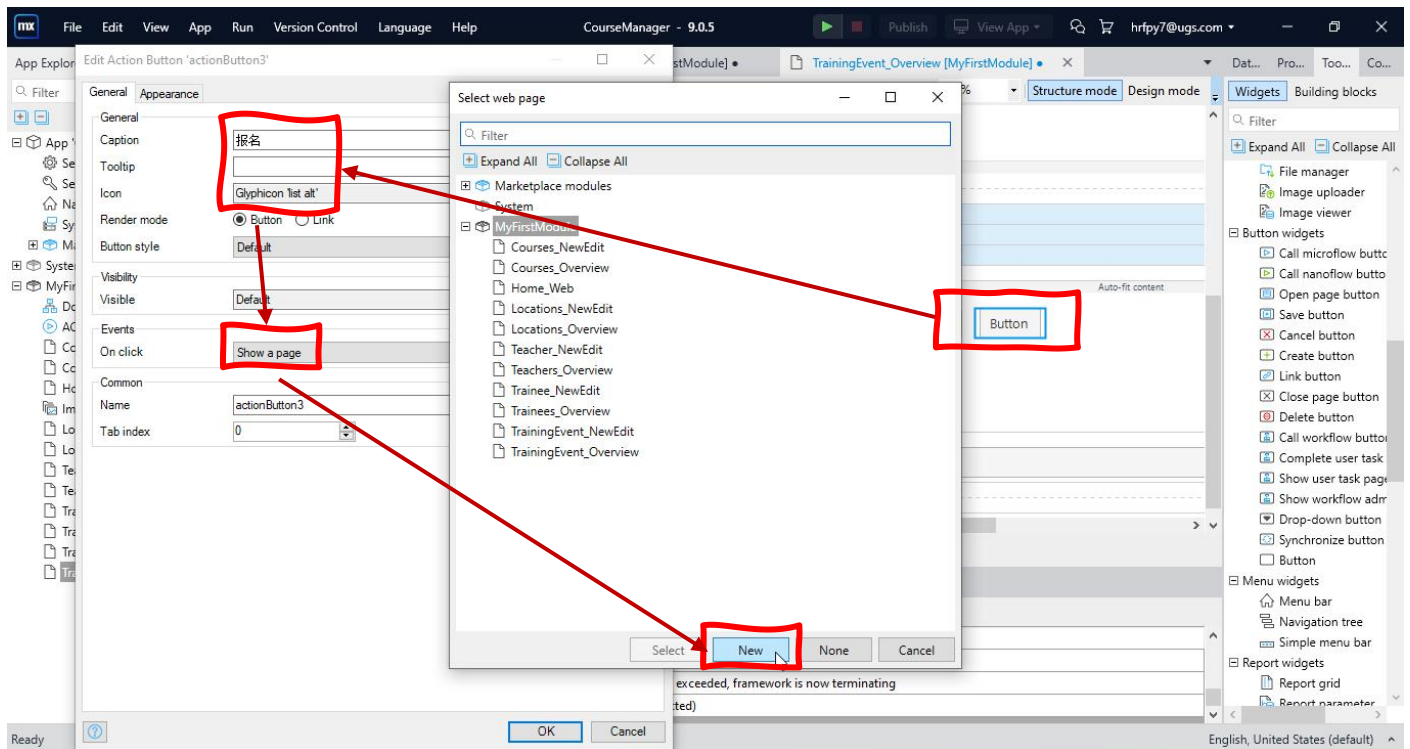


2.3 报名新增

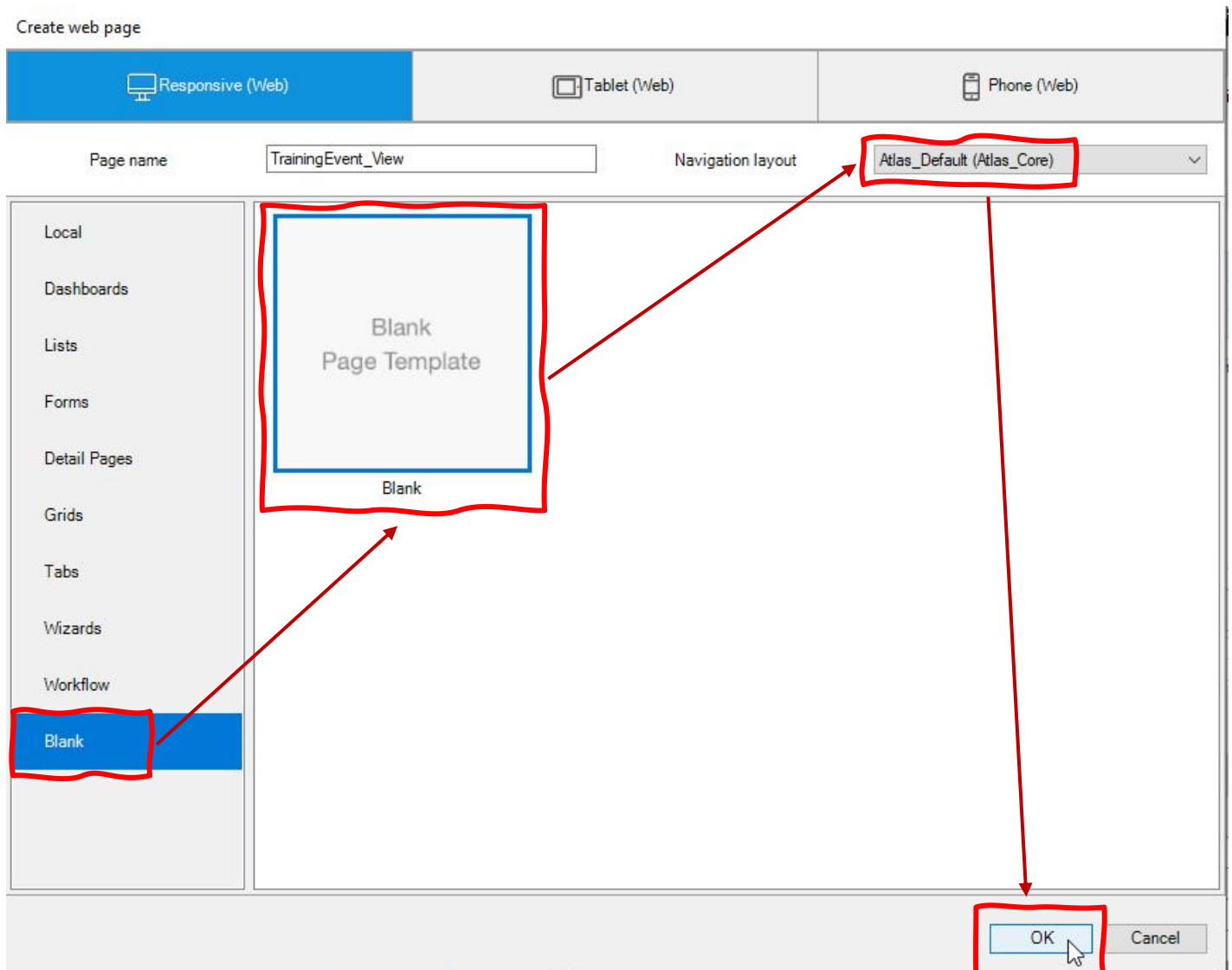
- 在活动列表中增加一个按钮，跳转到需求中的报名页面。双击目录树中的活动列表 **TrainingEvent_Overview** 页面，然后在 **3 右** 搜索 button，拖入一个空白 button，然后双击。



2. 在弹出的 Edit Action Button (编辑按钮) 窗口修改 **Caption** (按钮的名称) 为**报名**。然后修改 **On click** 下拉框选择 **Show a page**(展示一个页面)，再在弹出的 Select Web Page(选择跳转页)页面选择 **New** 按钮。

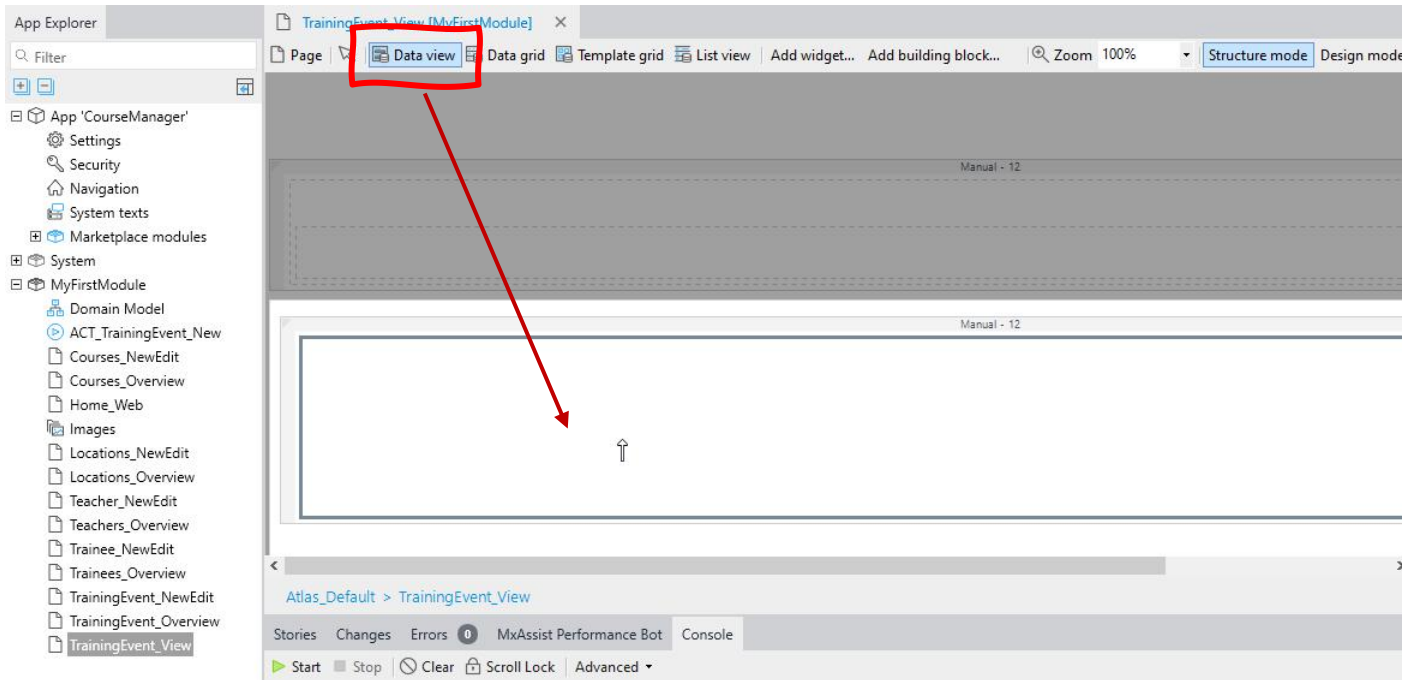


3. 在新弹出的窗口中，选择 **Blank** 空白模版，**Page name** 保留系统默认生成的名字，**Navigation layout** 选择 **Atlas Default**。然后点击 **Ok->Ok** 按钮，完成页面创建。

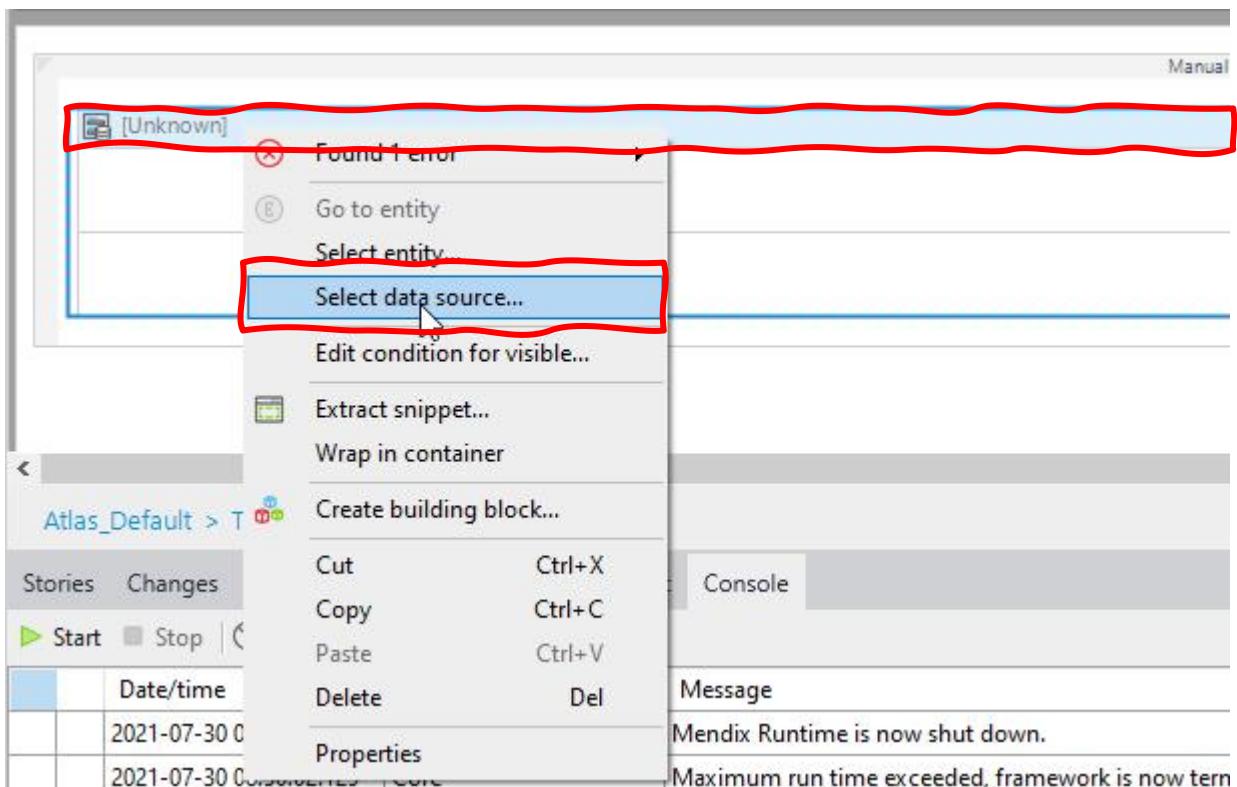


2.4 页面模型

1. 在目录树双击新页面 **TrainingEvent_View**，再菜单栏中选中 **Data view** 拖入到我们的页面中。



2. 右击拖入的 **Data view**，选择 **Select data source**



3. 在弹出的页面选择 **Select**，然后在弹出的页面上在选择 **TrainingEvent** 活动数据源，然后点击 **Select->Ok** 按钮。之后在弹出的页面选择 **NO** 按钮（不自适应）。

Select Data Source

Type ☒ Context ☐ Microflow ☐ Nanoflow ☐ Listen to widget

Entity (path) (none) Select...

? OK Cancel

Select Entity

Filter

+ Expand All - Collapse All

☒ Marketplace modules

☒ System

☒ MyFirstModule

- ☒ Courses
- ☒ Location
- ☒ Registration
- ☒ Teacher
- ☒ Trainee
- ☒ **TrainingEvent**

Select None Cancel

Select Data Source

Type ☒ Context ☐ Microflow ☐ Nanoflow ☐ Listen to widget

Entity (path) MyFirstModule.TrainingEvent Select...

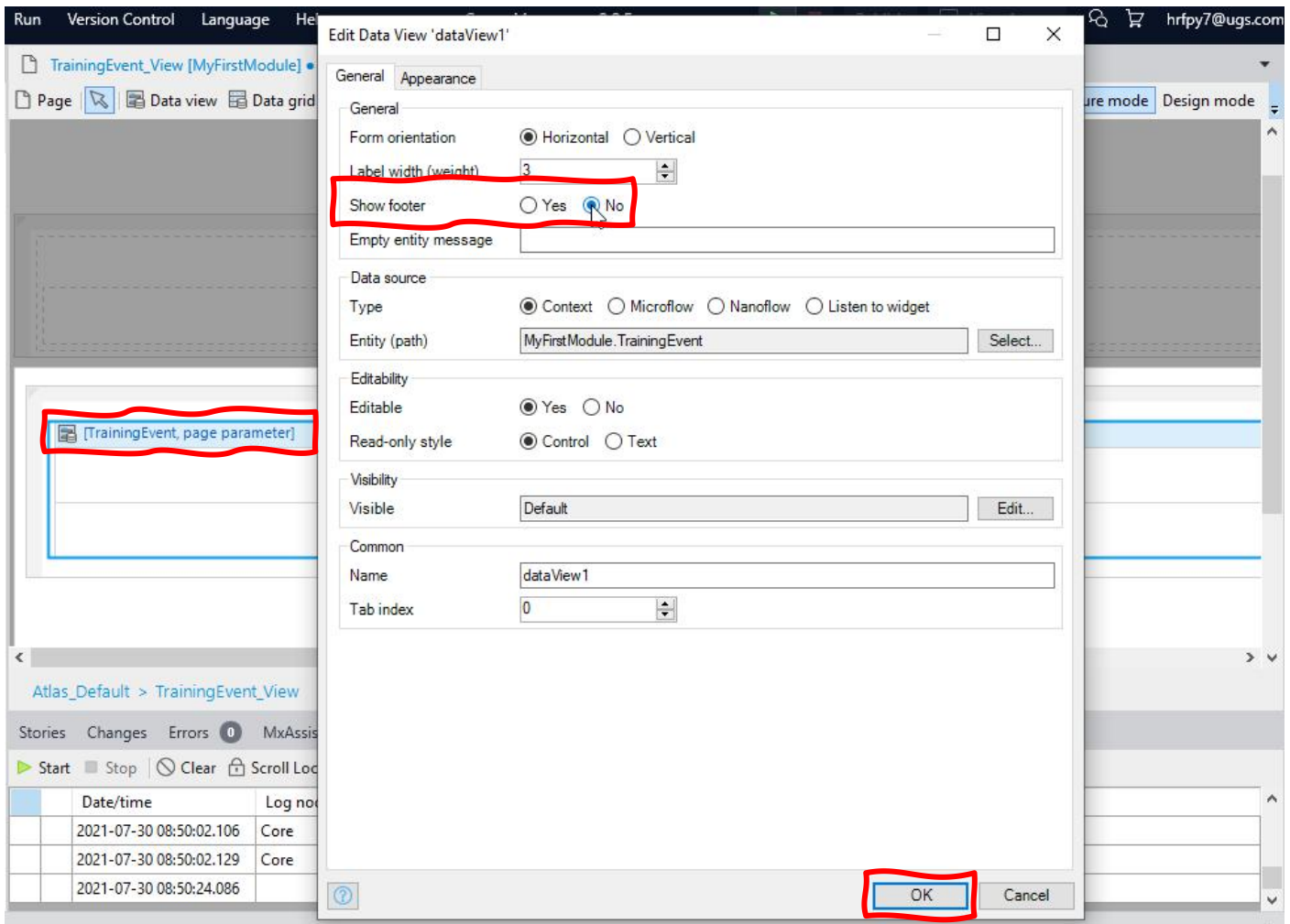
? OK Cancel

Question

? Do you want to automatically fill the contents of the data view?

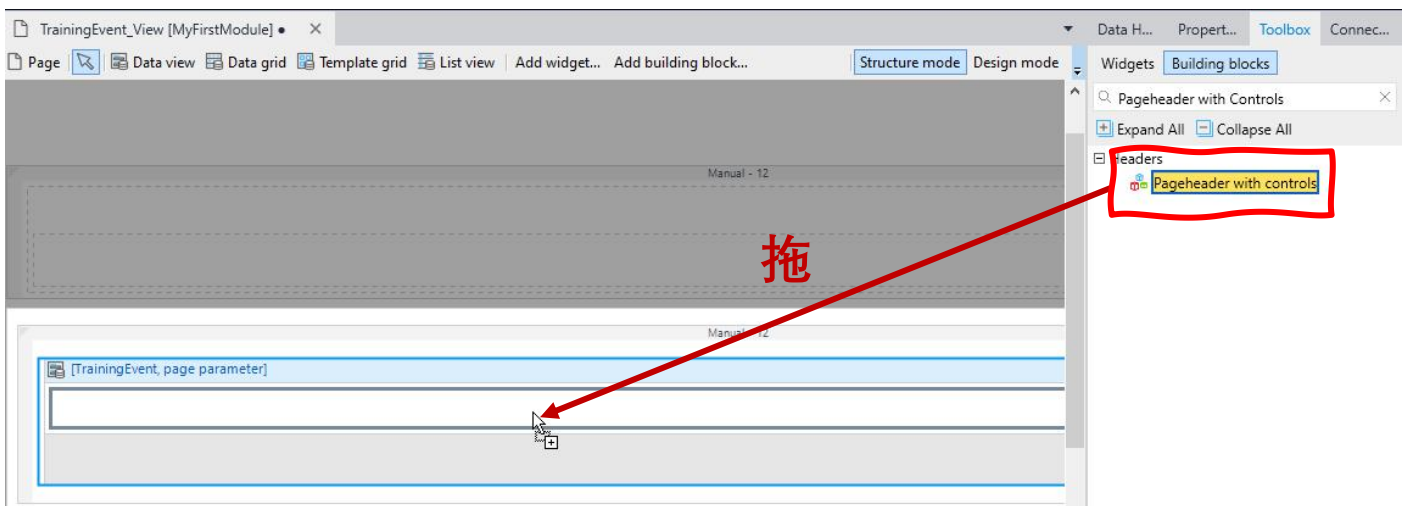
Yes No

4. 双击 **Data view** 数据表格，然后将 **Show footer** 改为 **No**，然后点击 **Ok** 按钮。这一步就是为好看一些。

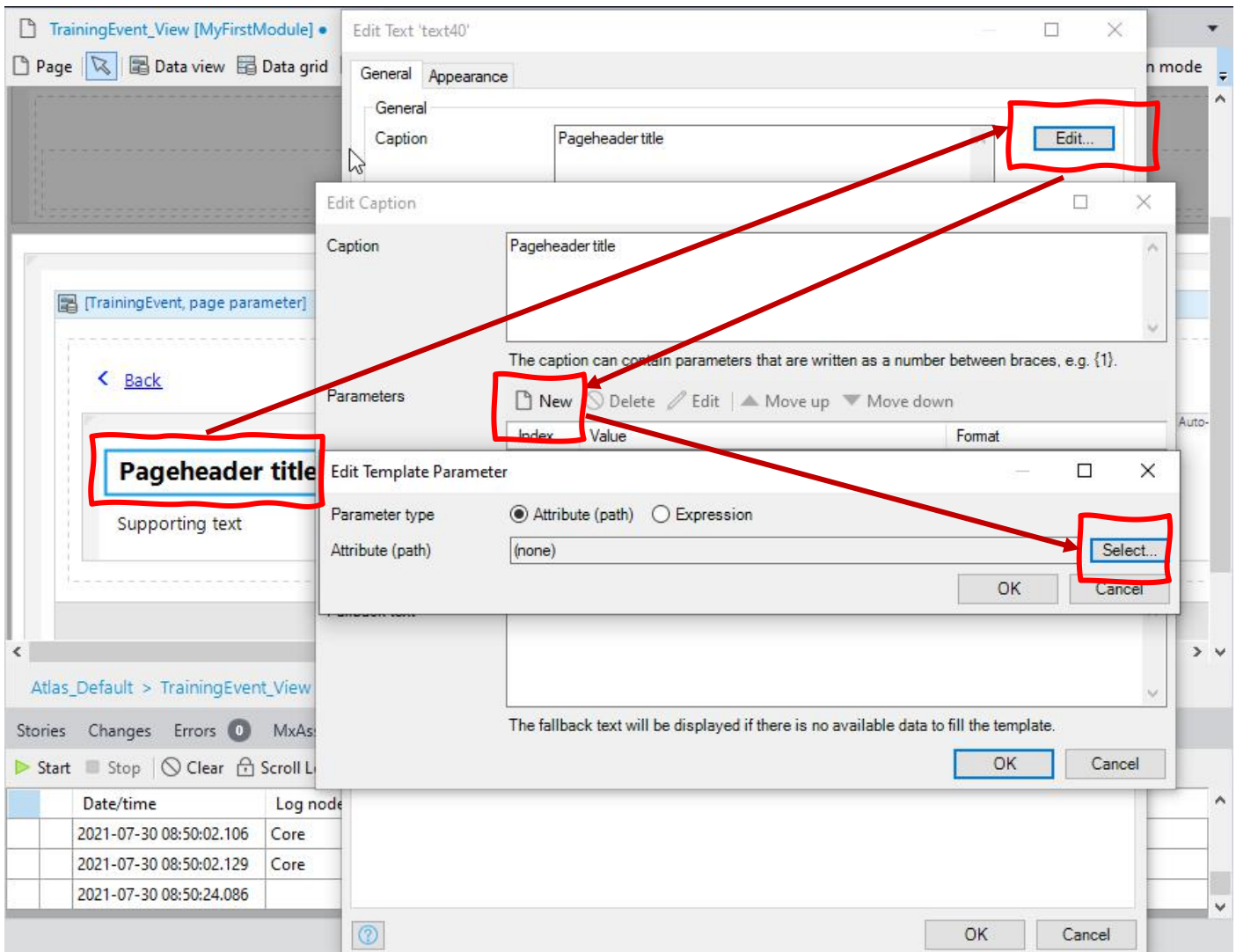


2.5 页面详情

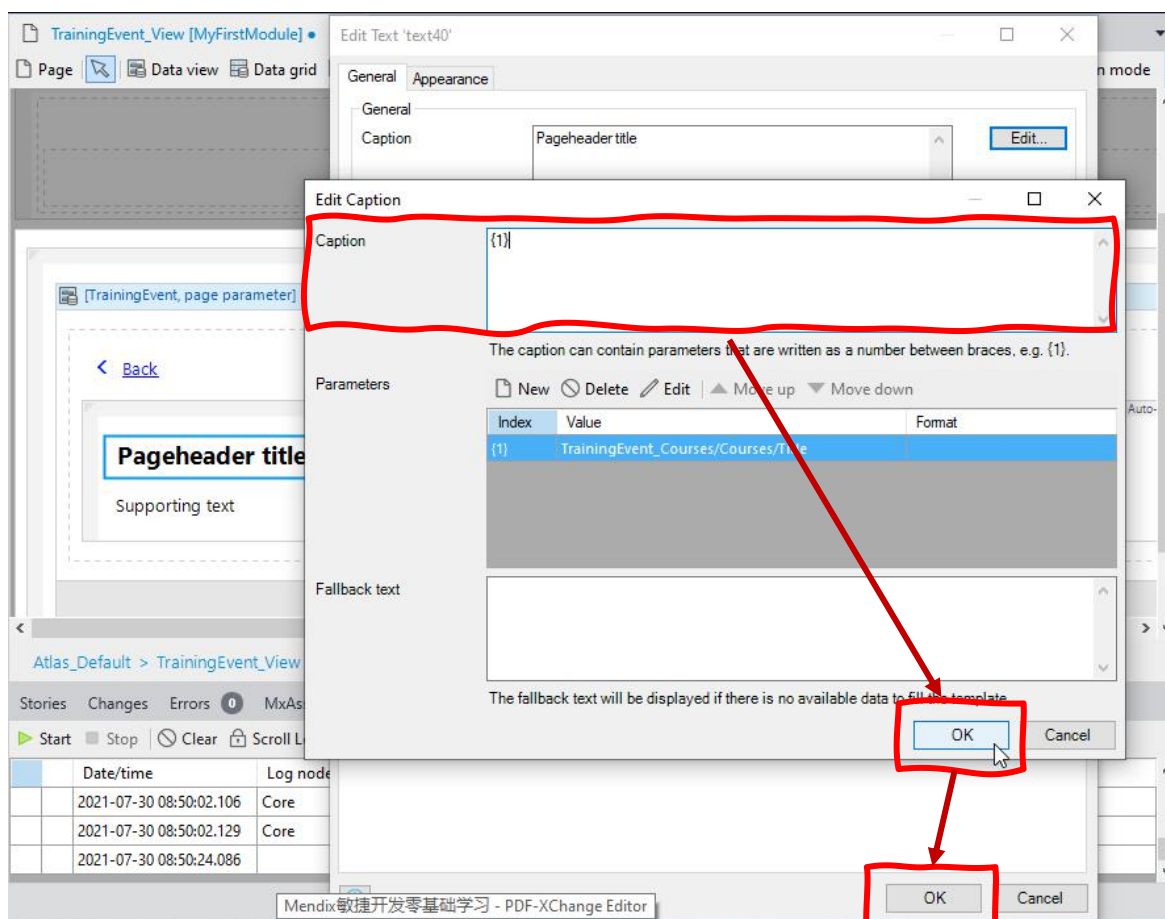
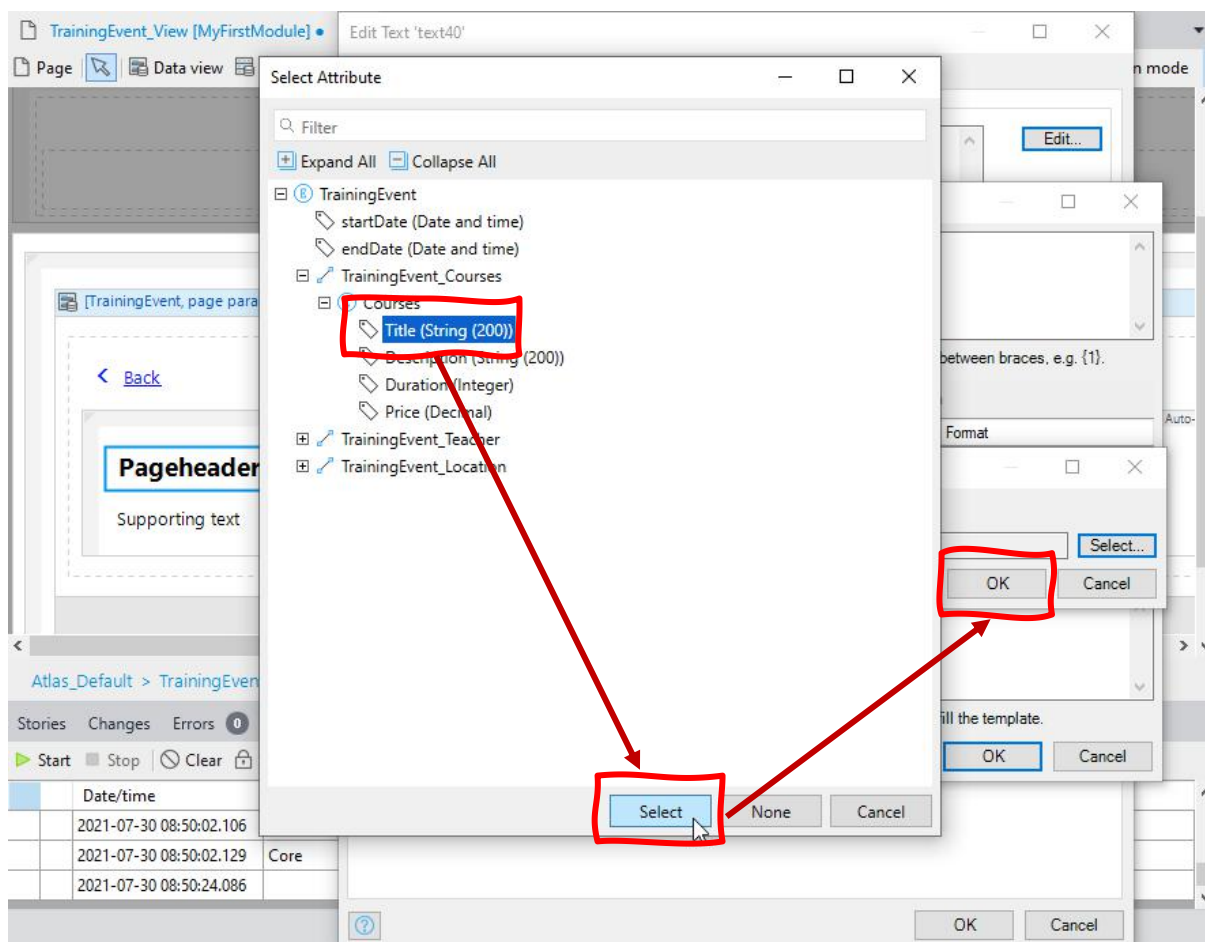
1. 然后在 **3 右** **Toolbox** 的 **Building blocks** 搜索 **Pageheader with controls** 将其拖入到 **Data view** 数据表格中。



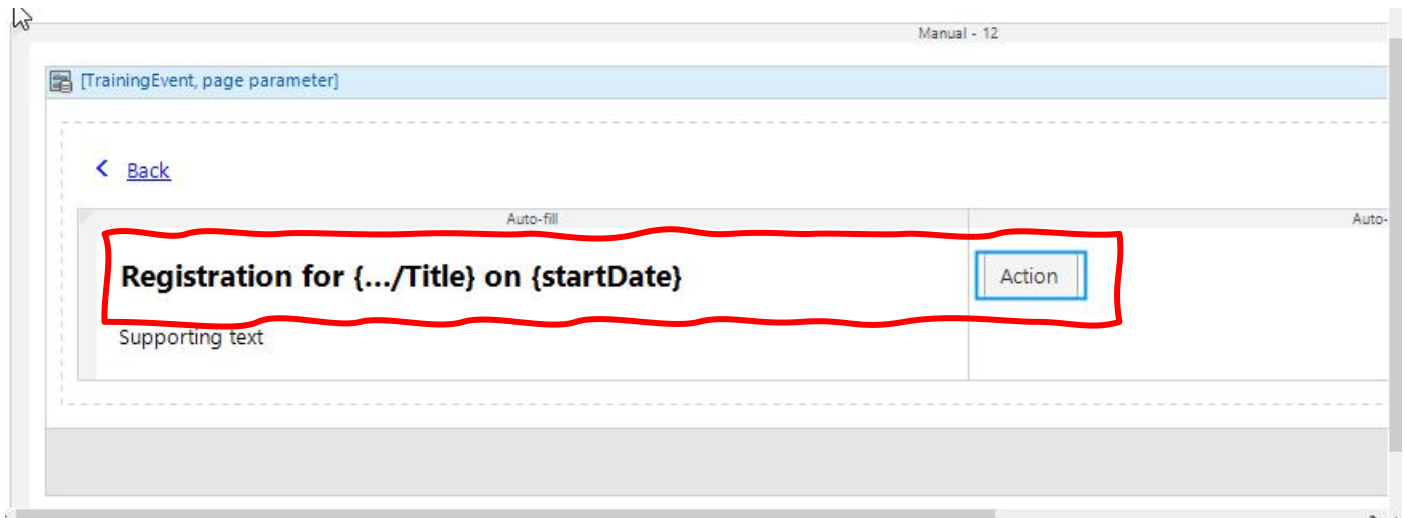
2. 双击 **Pageheader title** 使文本显示课程标题。在弹出的 **Edit Caption** 编辑文本页面点击 **Edit** 按钮，再点击 **New** 按钮，选择 **Select**,



3. 在弹出的窗口选择课程的标题 Title 属性，然后点击 Select->Ok 按钮。然后将序号为{1}的数据进行回显，按照下图修改 Caption 为{1}即可，然后点击 Ok->Ok 按钮

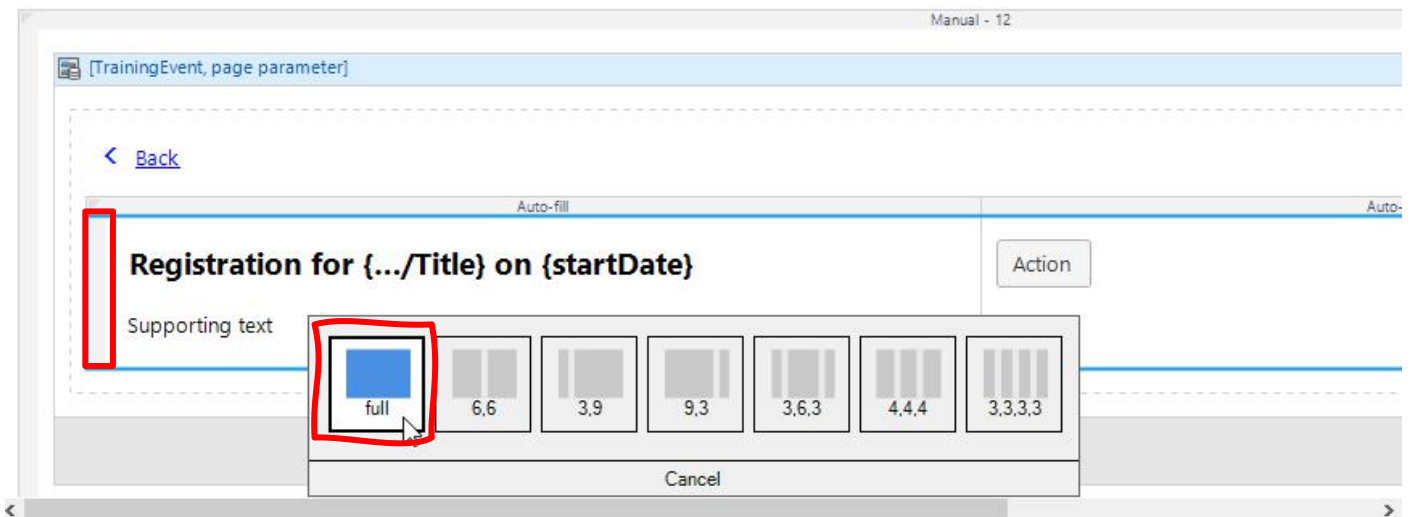


- 可以根据已学的知识稍微标明一下页面，并删除一个按钮。

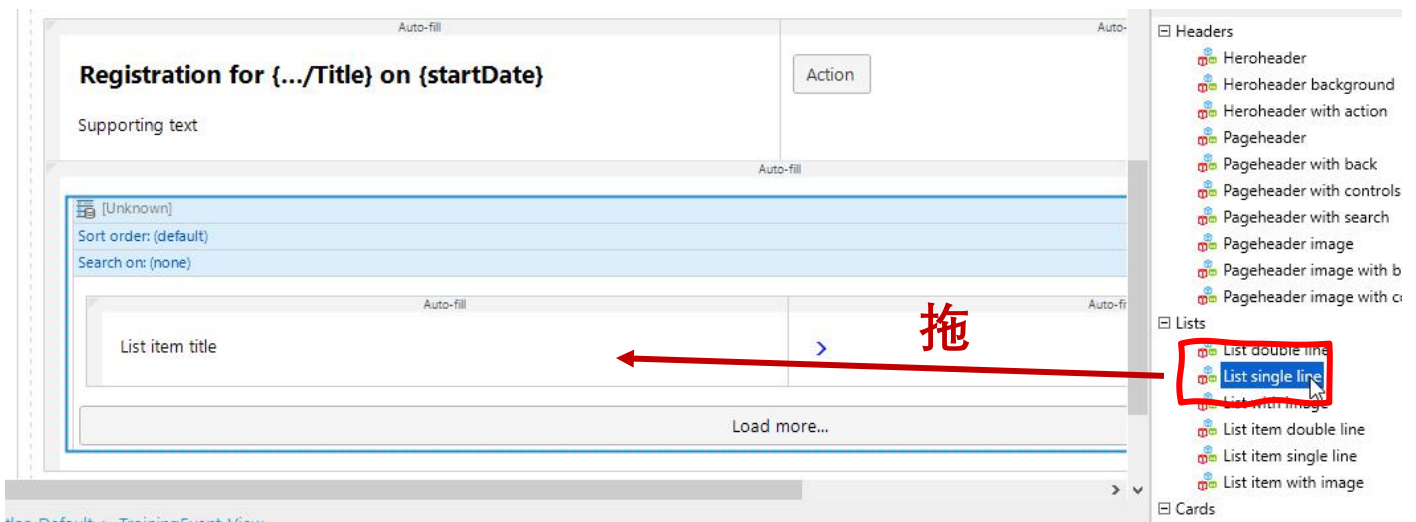


2.6 页面列表

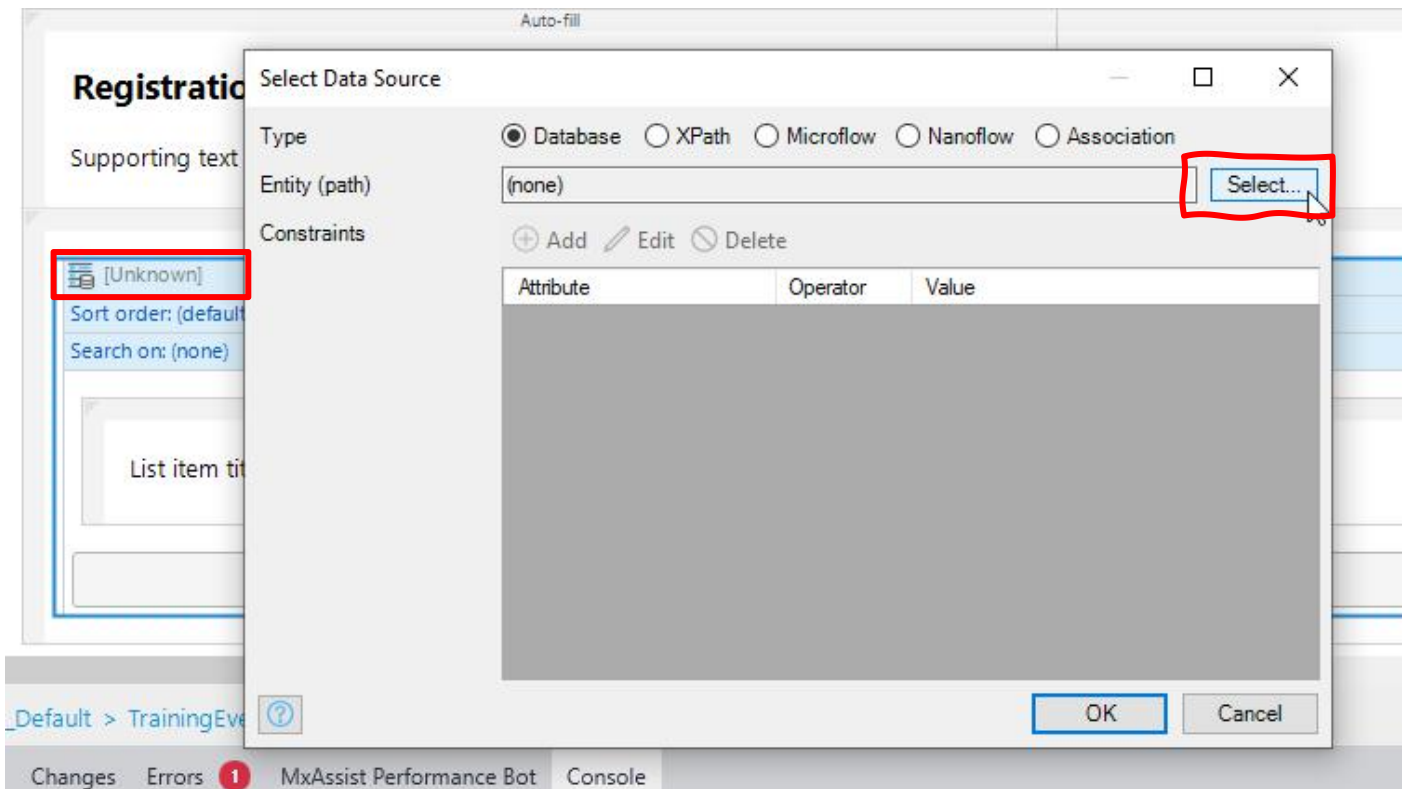
- 在 **TrainingEvent** 数据源内中添加一行：右键左边灰色处选择 **Insert row below**，选择 Full 一列。



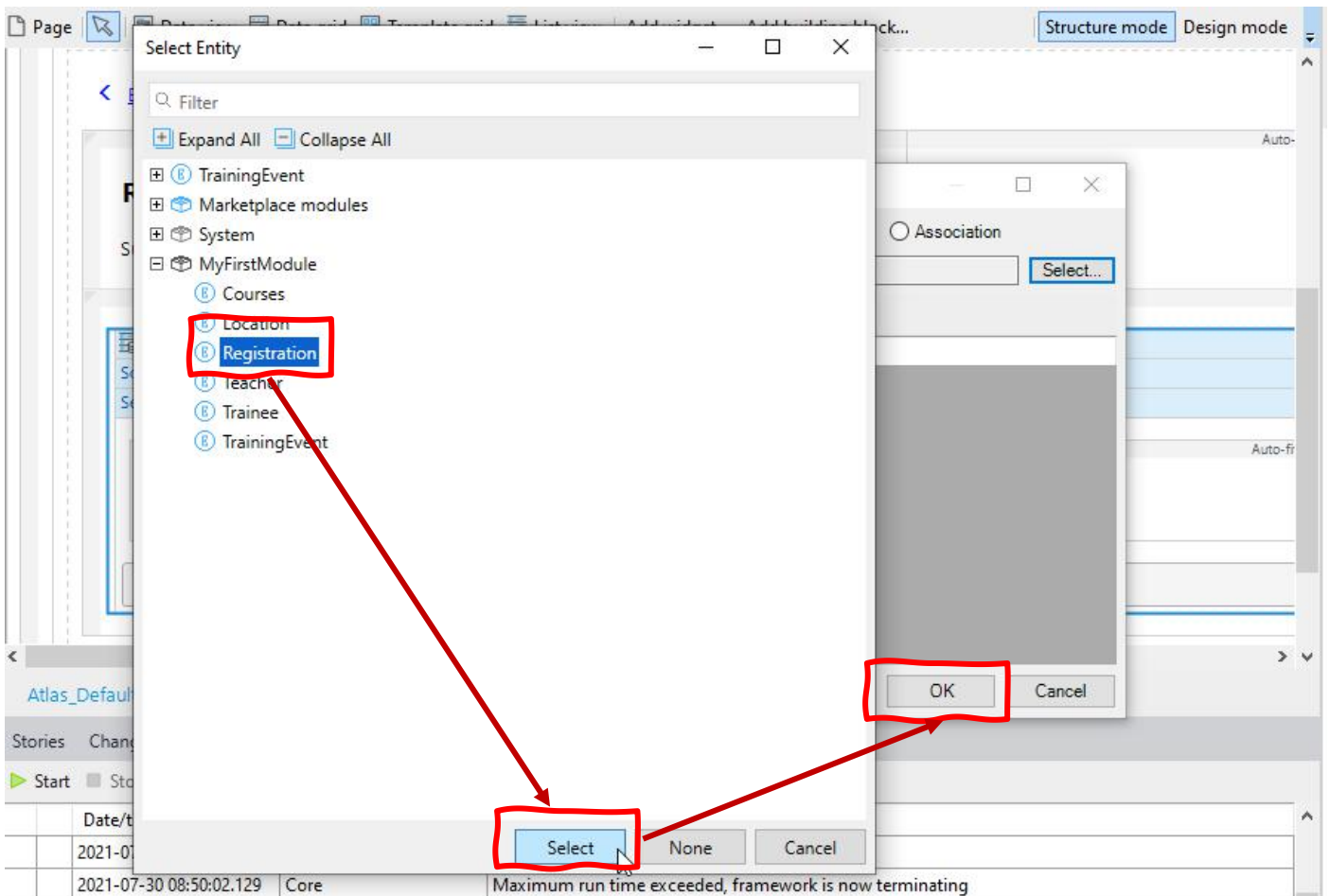
- 在 **3右**区 **Toolbox** 的 Building **blocks** 搜索 **list**，将 list2 拖到新增的那一列中。



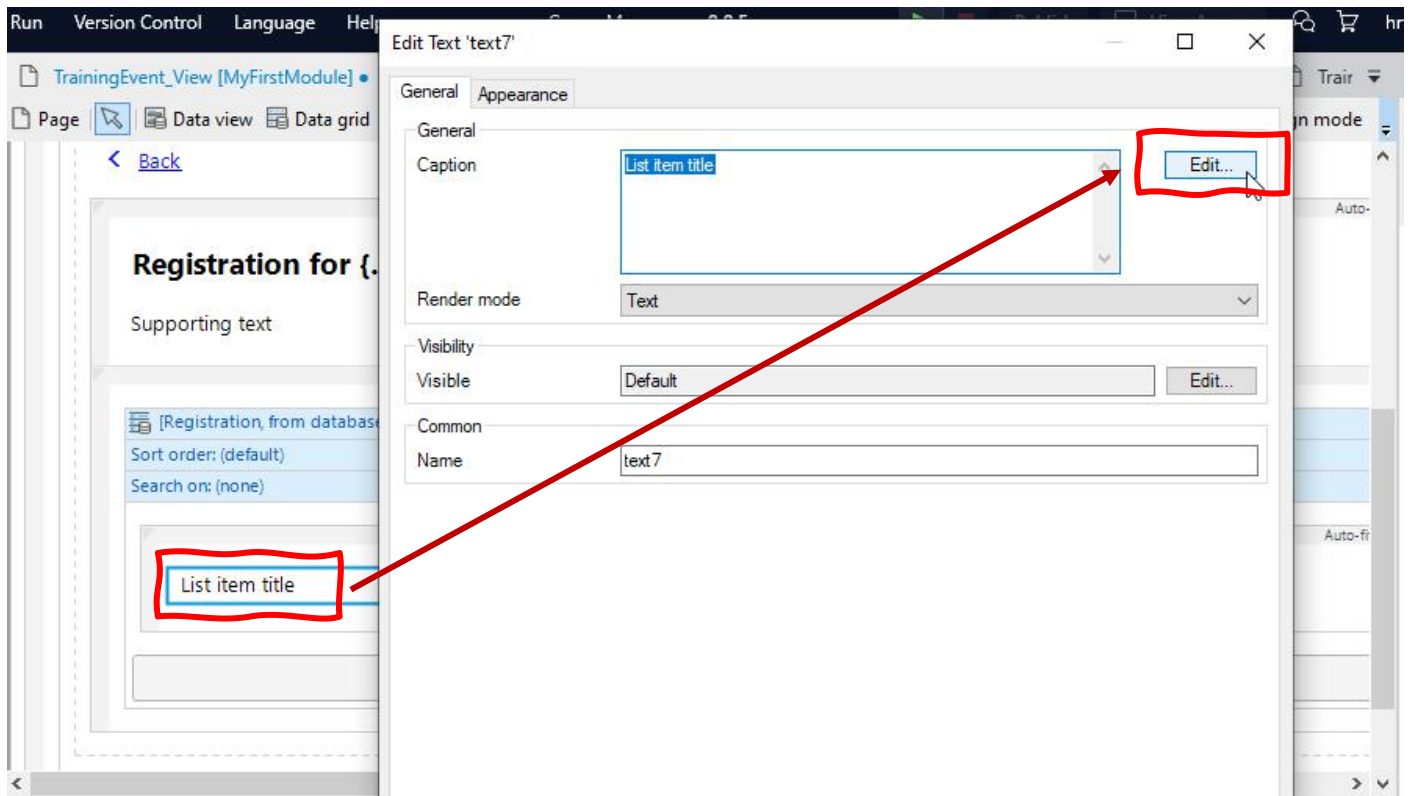
3. 右击刚拖入的 **List**，点 **Select data source** 选择数据源。在弹出的窗口选择 **Select** 按钮。



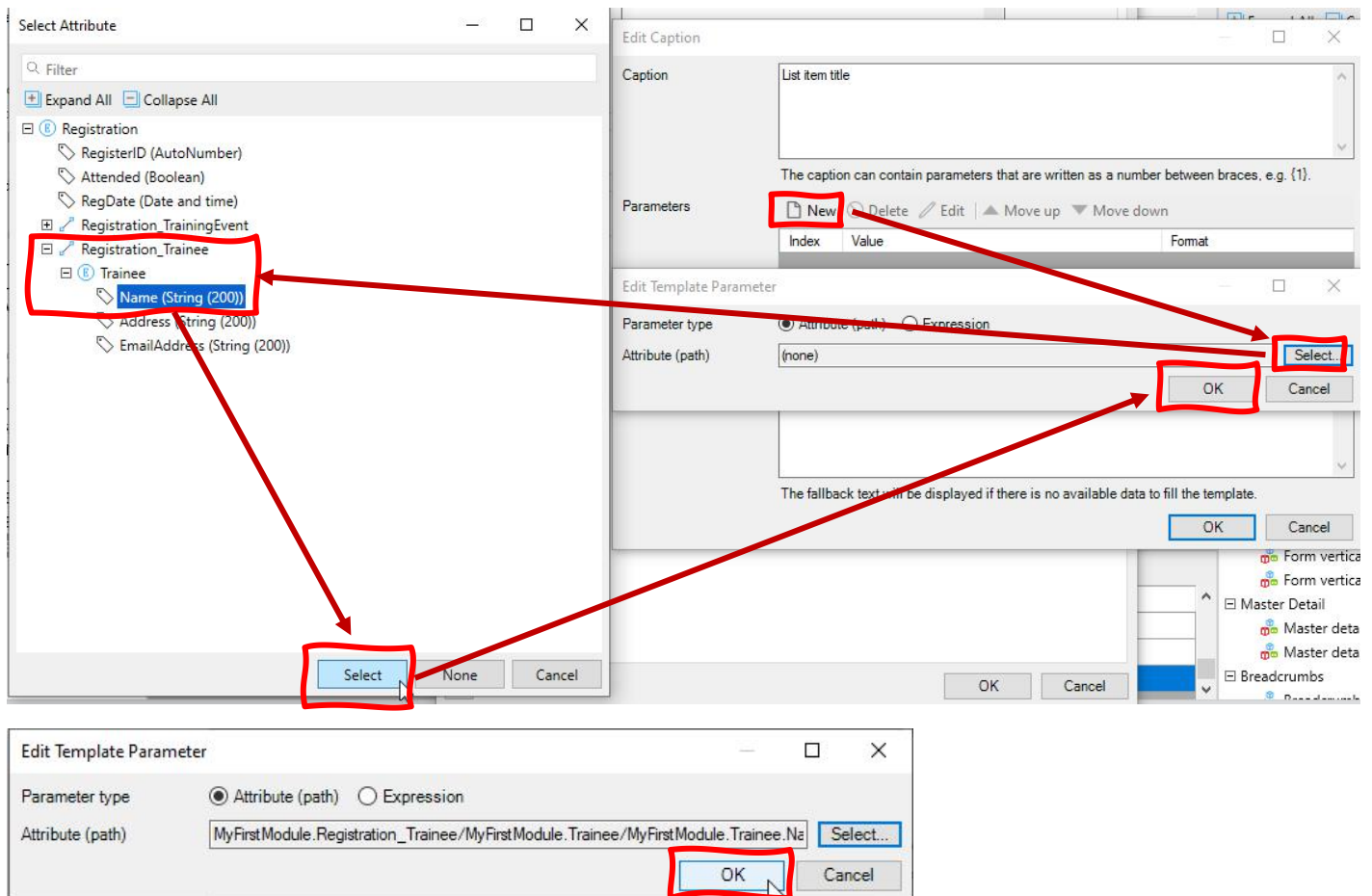
4. 选择 **Registration**(报名人员)，然后点击 **Select->Ok** 按钮完成选择。



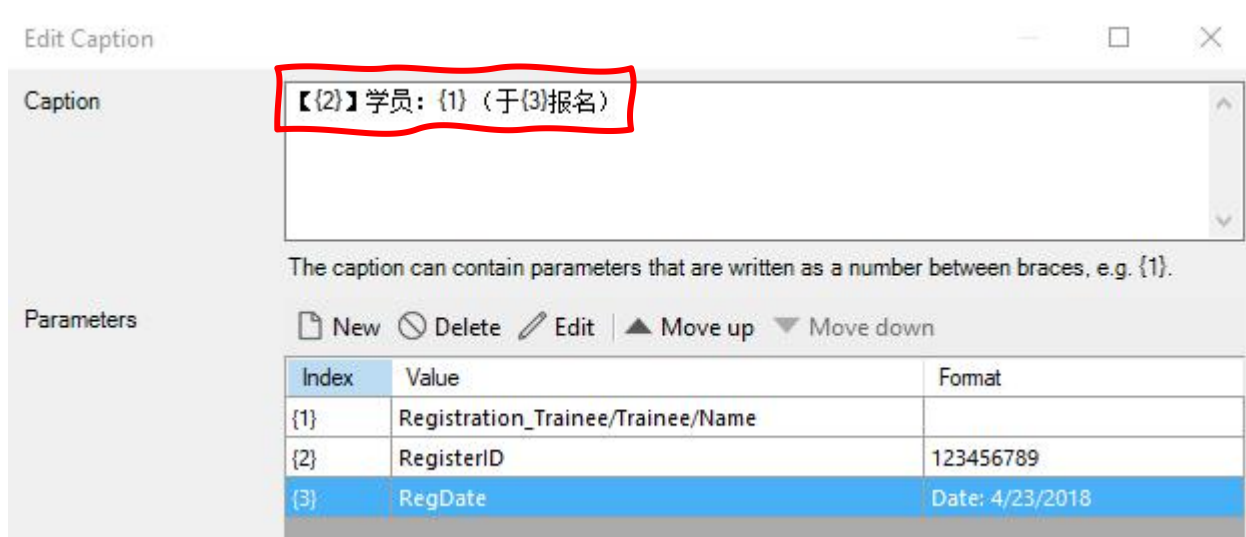
5. 接下来修改列表项的显示，双击 **List item title**，然后在弹窗 **Caption** 处选择 **Edit**。



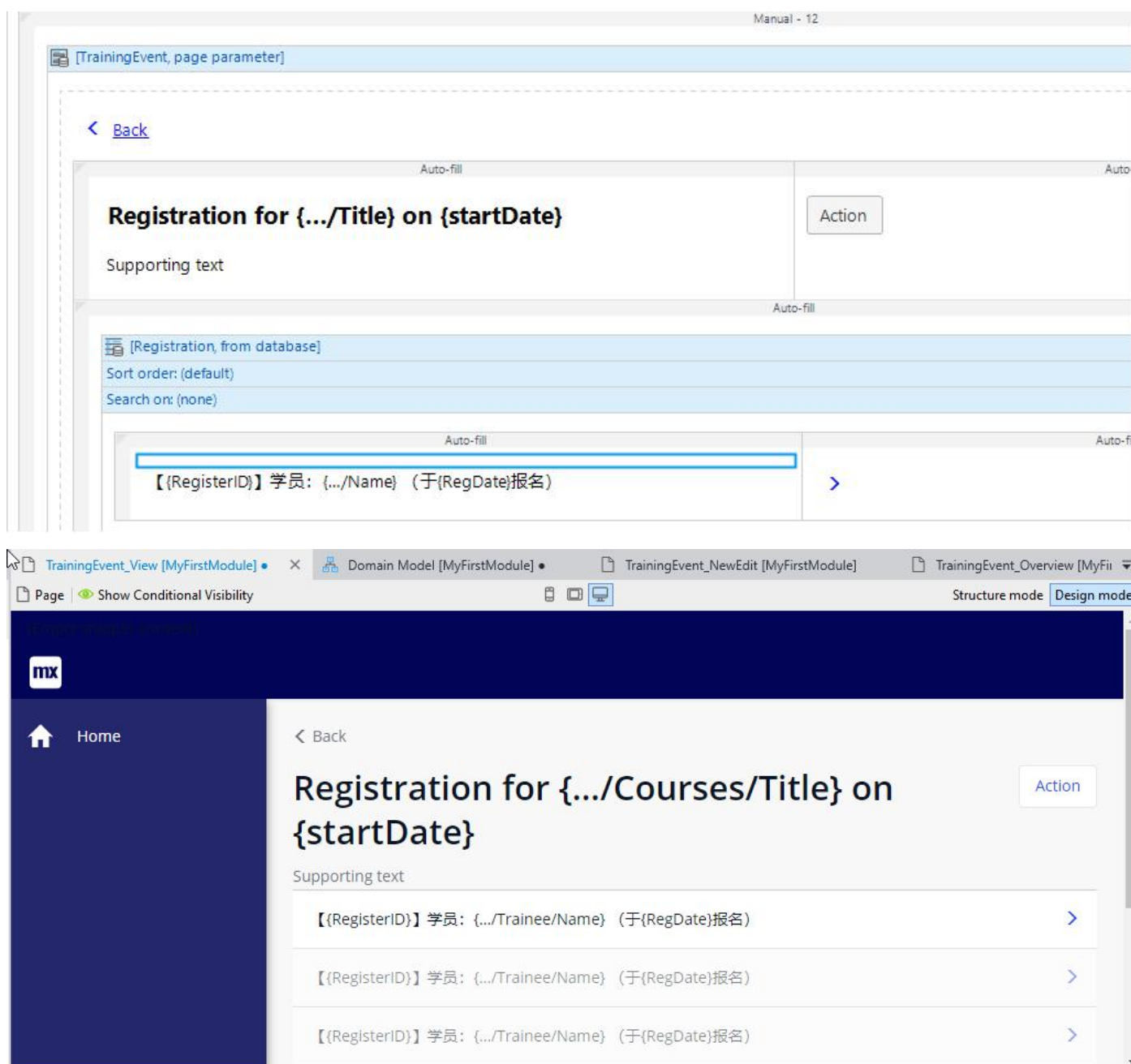
6. 在 **Parameter** 参数处选择 **New** 按钮，再选择 **Select** 按钮。在弹窗选择 **Registration_Trainee** 中的 **Name** 报名人员。然后点击 **Select->Ok** 按钮。



7. 用同样的方式添加 **RegisterID** 和 **RegDate**，并按下图修改 **Caption**，点击 **Ok->Ok** 按钮。

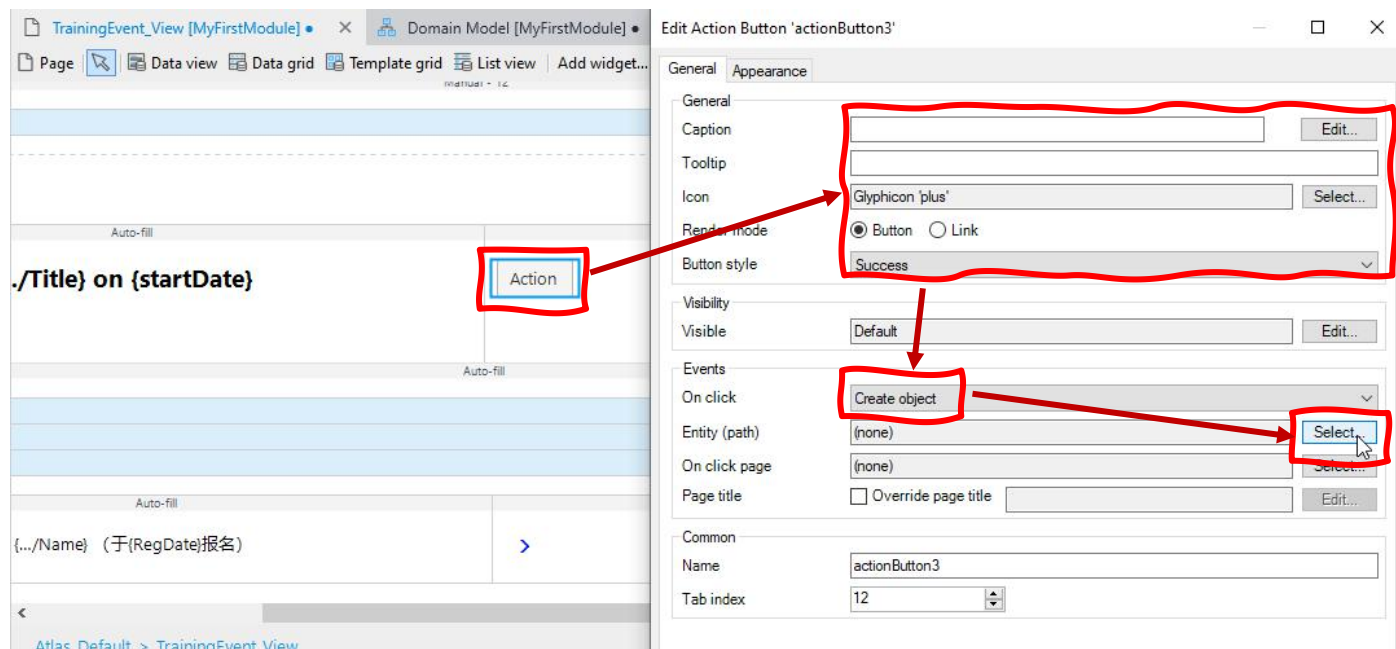


8. 页面的渲染就完成了，最终效果如下：



2.7 列表新增

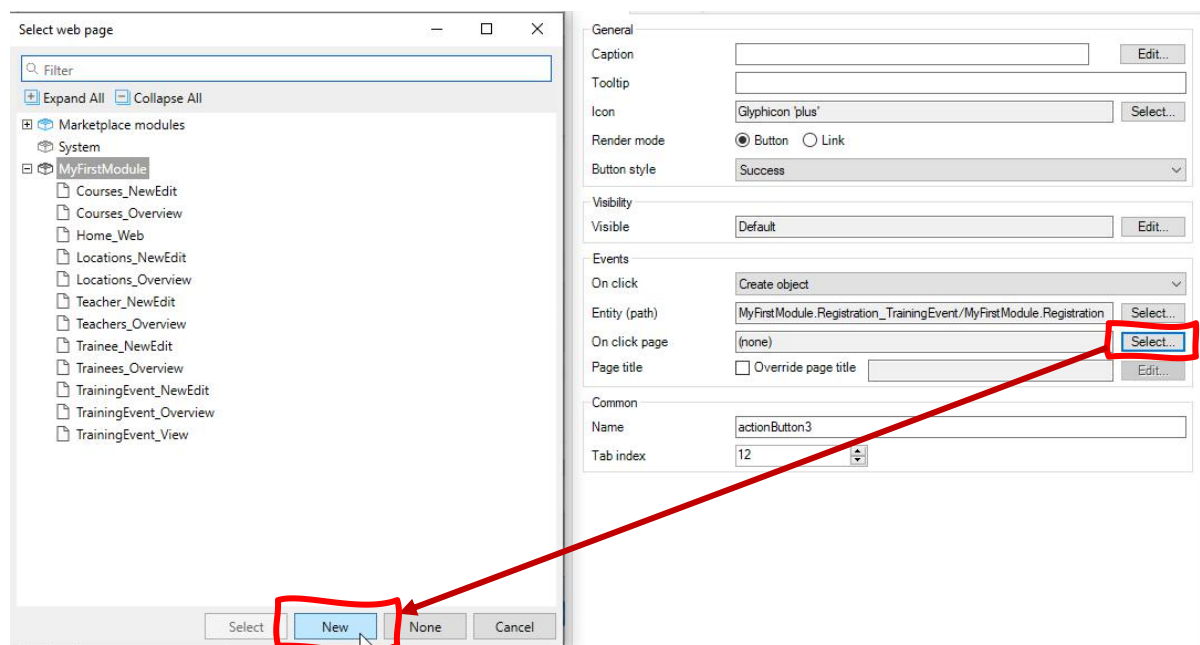
1. 接下来**新增报名人员**：我们双击页面标题边上的 **Action** 按钮，如下图配置按钮样式，并在 **On click** 下拉框中选择 **Create object** 选项，在新显示的 **Entity(Path)**中点击 Select 按钮。



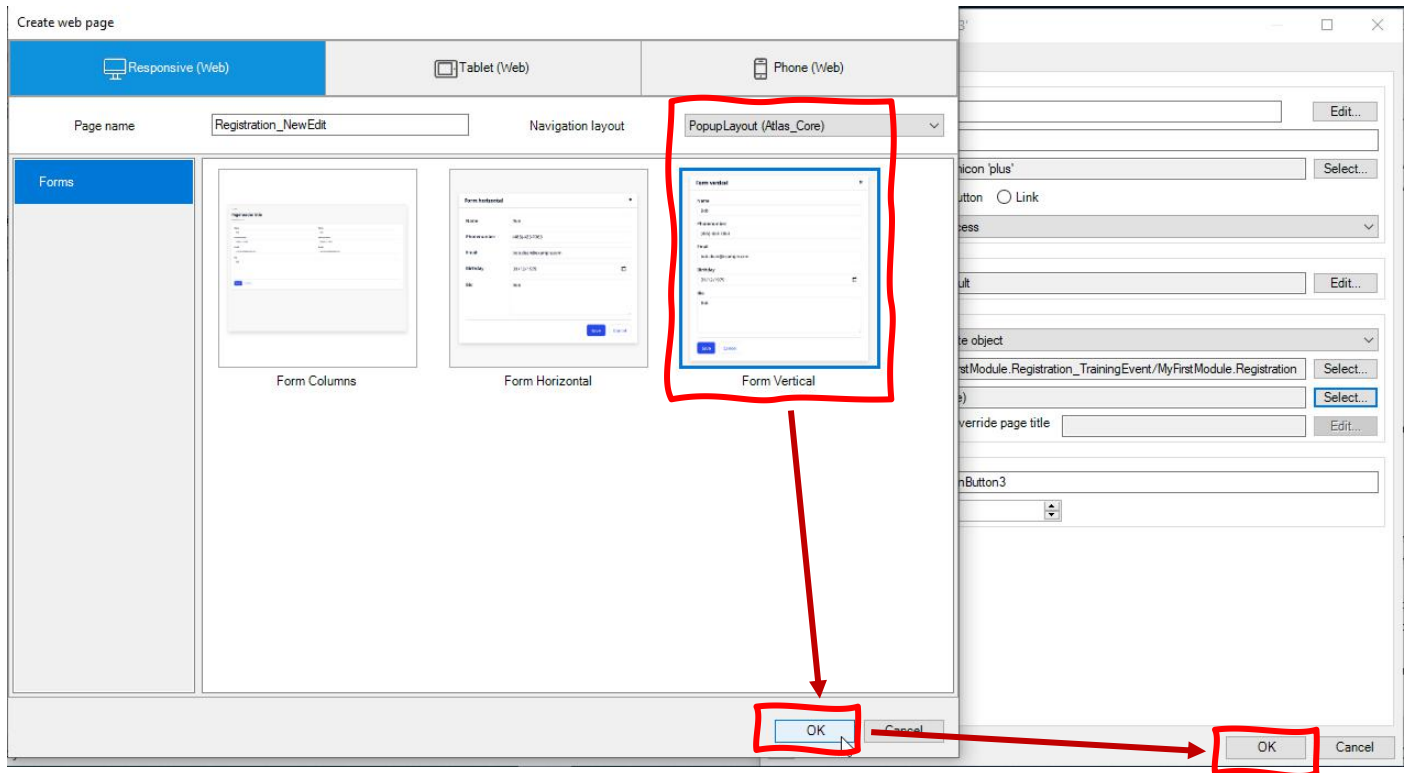
2. 在弹出的 **Select Entity** 页面选择与 **TrainingEvent** 关联的 **Registration**，然后点击 Select 按钮。



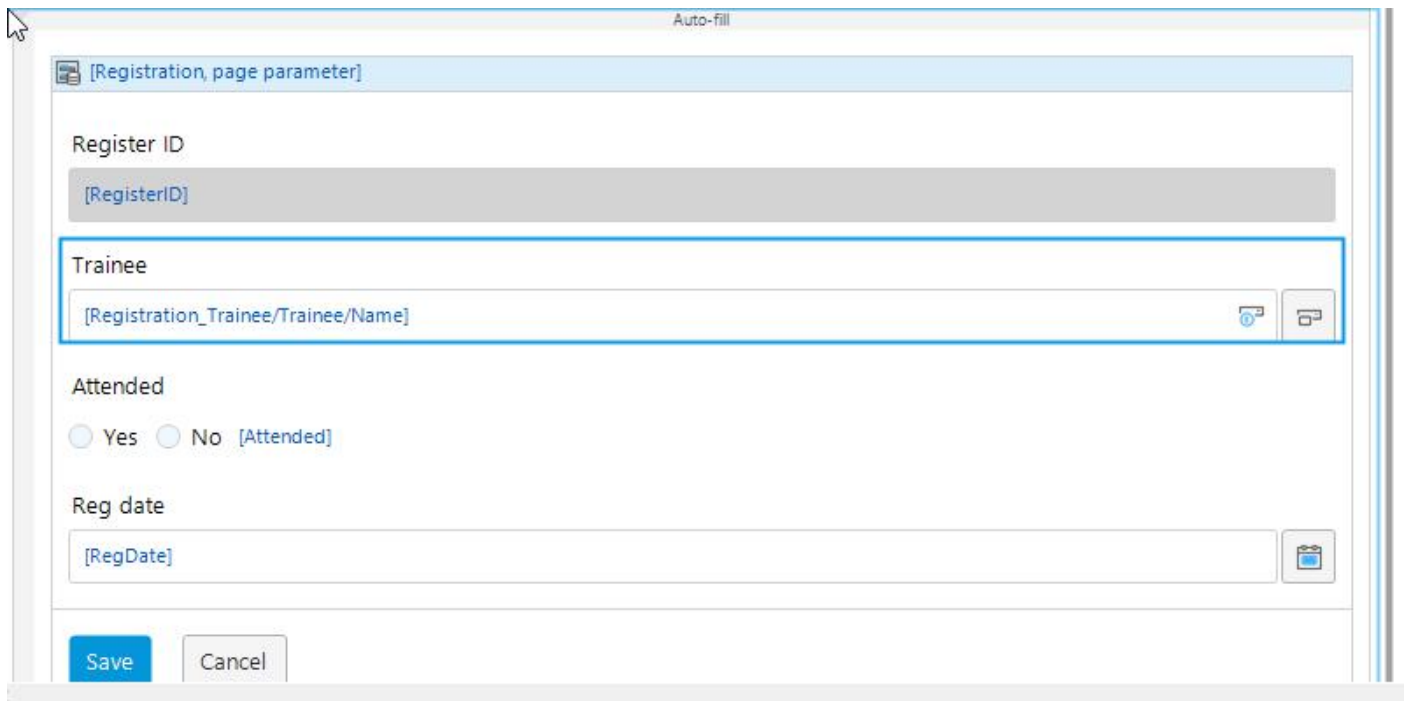
3. 接着 **On click page** 中点击 **Select** 按钮，在弹出窗口选择 **New**。



4. 新页面中选择 **Form Vertical** 模版, **Navigation layout** 选择 **PopupLayout**, 再点击 **Ok->Ok** 按钮。

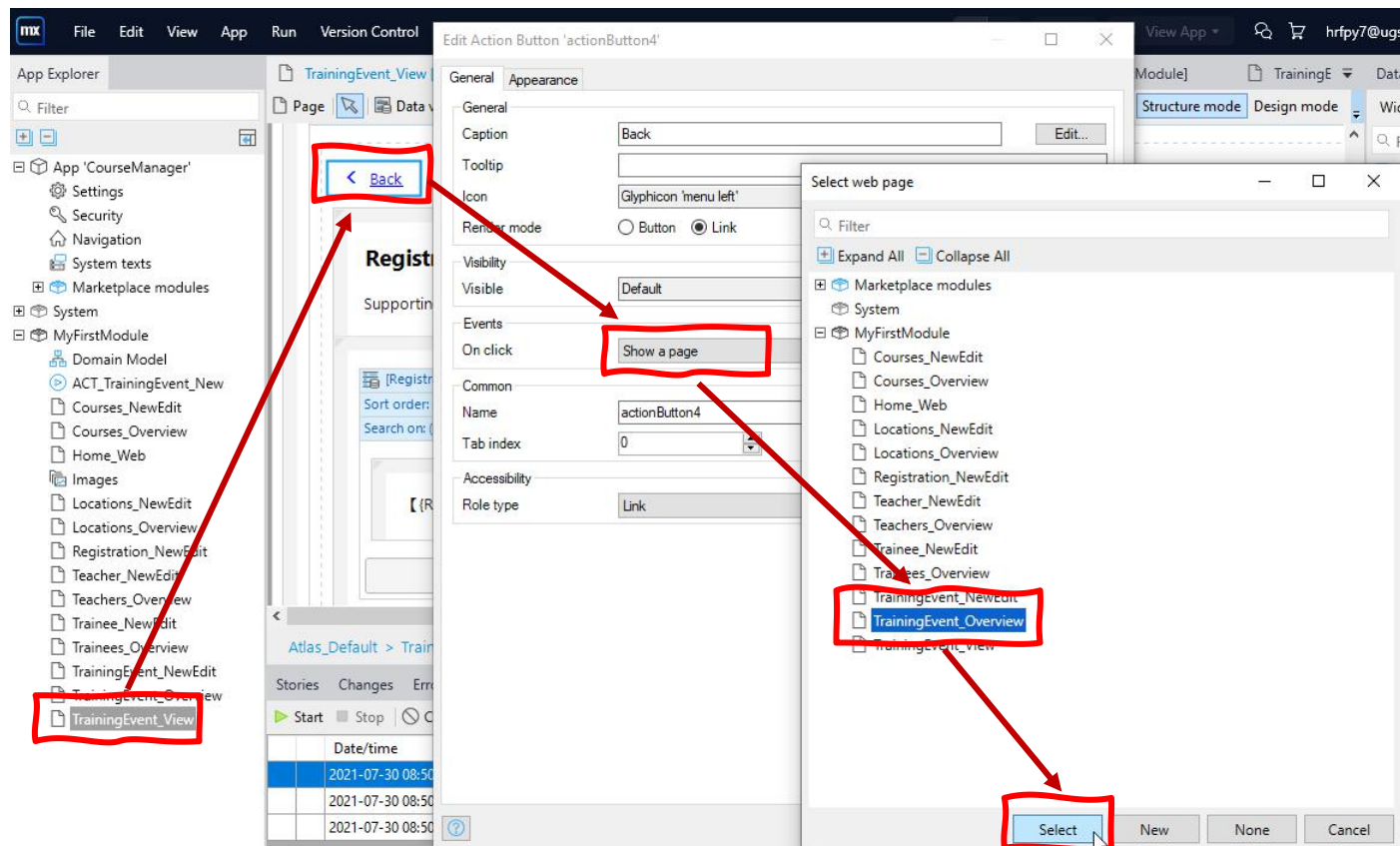


5. 双击目录树的 **Registration_NewEdit** 调整选项列的顺序, 新增报名条目的功能就完成了。



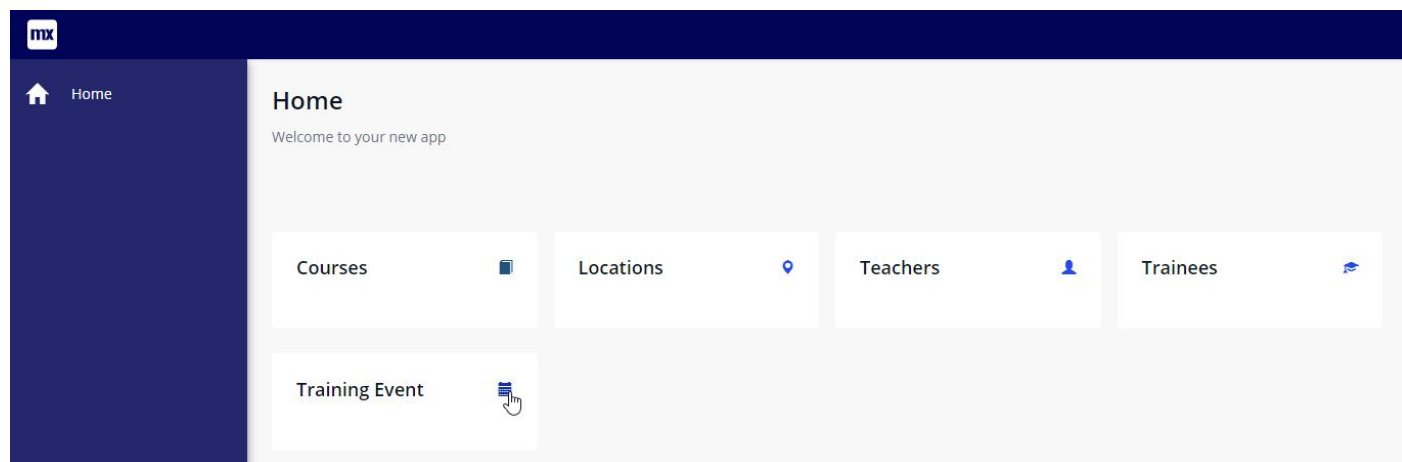
2.8 返回活动列表页

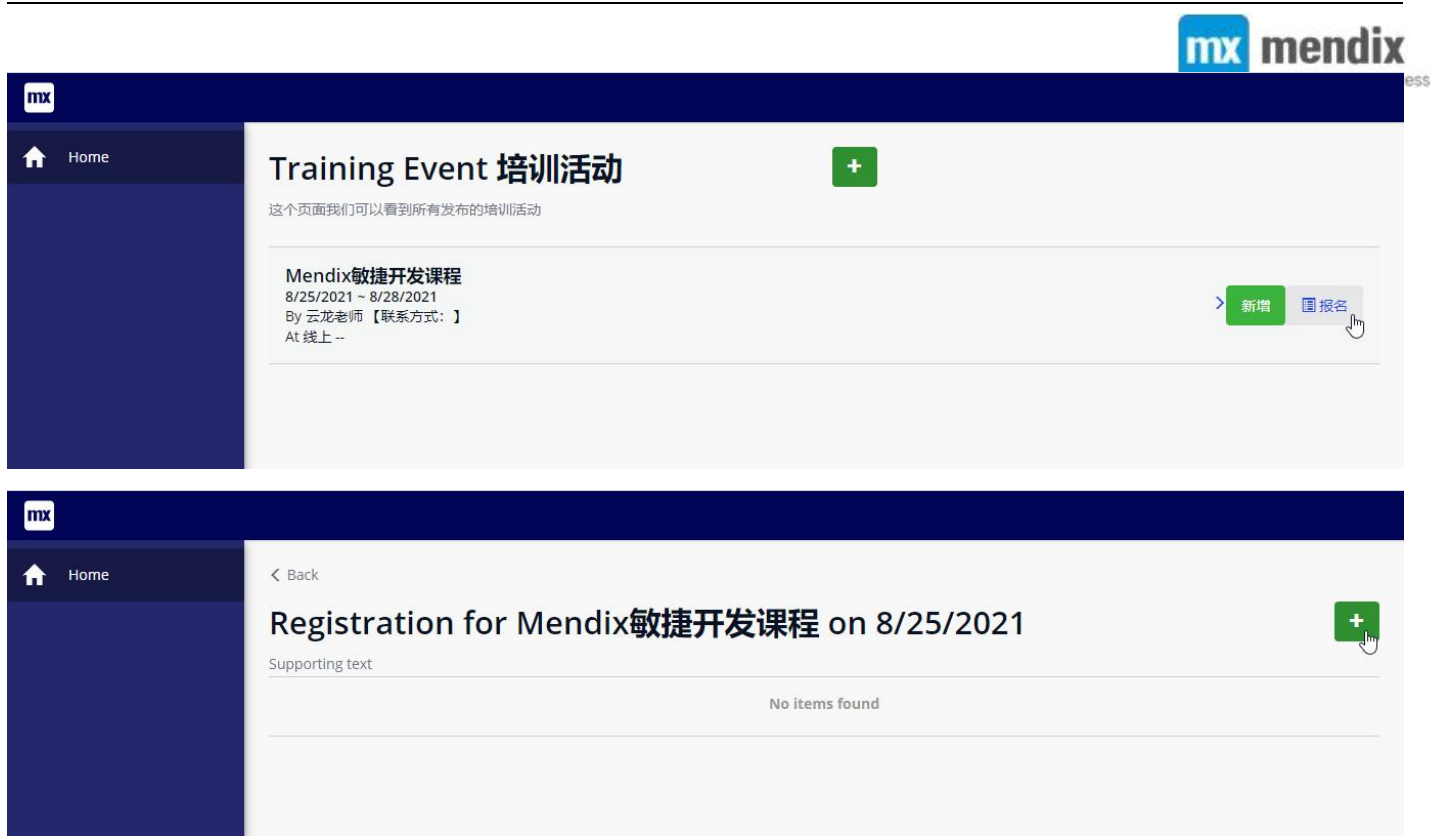
1. 回到 TrainingEvent_View 页面，双击页面左上的 **Back** 按钮，在新窗口 On click 处选择 Show a page，在 Select web page 窗口选择 TrainingEvent_Overview（回到活动列表页面），再点击 Select->Ok 按钮。



2.9 效果展示

1. 运行项目看效果。点击 **TrainingEvent** 卡片，点击**报名**，再点击 **Add** 按钮进行新增。

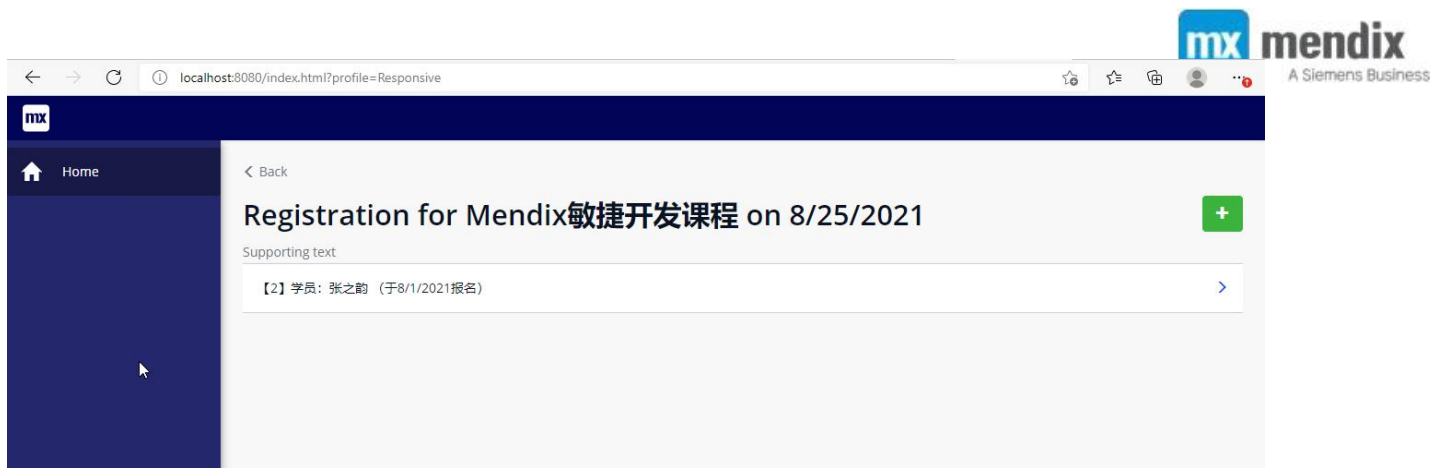




2. 输入报名人信息，点击 Save 保存按钮，就可以在列表中发现新增的数据。（Register ID 通过每次点击 add 新增报名计数，即使没有保存下一次点击 add，Register ID 依旧会+1）

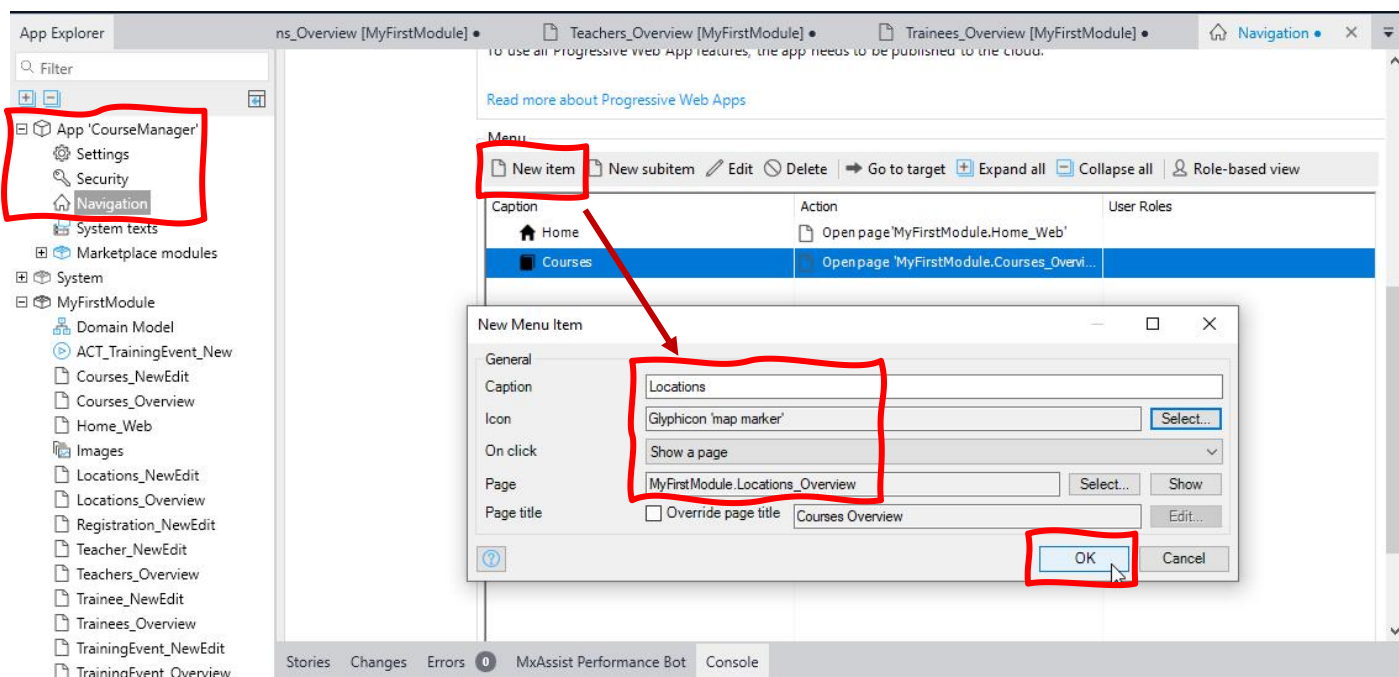
The screenshot shows the 'Edit Registration' form. The form has a title bar with 'Edit Registration' and a close button (X). The form contains the following fields and controls:

- Register ID:** A text input field containing the value '2'.
- Trainee:** A dropdown menu with the selected value '张之韵' (Zhang Zhiyun).
- Attended:** A radio button group with two options: 'Yes' (unselected) and 'No' (selected).
- Reg date:** A date input field containing the value '8/1/2021'.
- Buttons:** At the bottom of the form, there are two buttons: 'Save' (highlighted with a mouse cursor) and 'Cancel'.



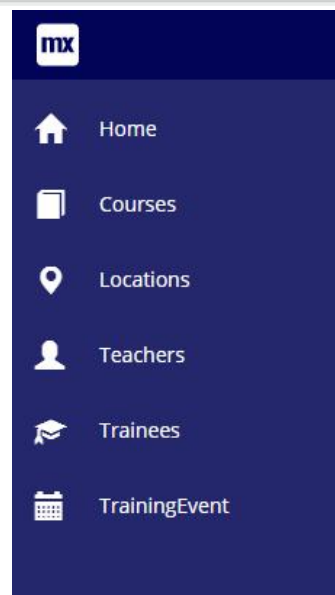
2.10 Nav Bar 设置

回到一、微流【3.2 举例 2】之前，在导航栏添加跳转页面的按钮：在目录树中双击 **App 'Course Manager'** 中的 **Navigation**，向下拉 **2 中** 页面的 Menu 处点击 New item 添加新的跳转选项，如下图。



名称	图标	页面命名
Courses	book	Courses_Overview
Locations	map	Locations_Overview
Teachers	user	Teachers_Overview
Trainees	education	Trainees_Overview
TrainingEvent	calendar	TrainingEvent_Overview

根据上表完成 Nav Bar 配置后，请跳转到一、微流【3.2 举例 2】继续。



四、版本管理

版本控制可以参考官网，比较详细。这里就不列举了。链接: [官方地址](#).

Mendix 敏捷开发零基础学习 《三》

数据删除保护机制、数据关联删除、**Security** 安全、调用外部接口、调用 **JAVA** 代码

- 该章节的例子是基于【[Mendix 敏捷开发零基础学习《一》](#)】和【[Mendix 敏捷开发零基础学习《二》](#)】的需求和例子来讲解。

目录

一、数据保护机制(*Prevention of Delete*)

1.业务需求

2.业务分析

3.项目实施

二、Mendix 权限(*Security*)

1、开启系统权限 **ProtoType/demo**

2、项目权限(**Project Security**)

3、模块权限(**Module Security**)

4、项目配置

5、开启系统权限 **Production**

三、用户新增

四、程序退出功能

五、调用 **JAVA** 代码

六、调用外部接口

一、 数据保护机制(Prevention of Delete)

1.业务需求

因为系统管理员有删除活动的权限，但是如果有人已经报名了，这时候删除，会导致数据丢失。所以甲方要求，如果有已经报名的人员，活动不能删除。

2.业务分析

这个需求在 Mendix 可以非常方便的实现。我给操作一下。

3.项目实施

- 我们双击 Domain Model 数据模型，双击下图所示的关系线，在弹出的页面我们可以看到 On delete of 'TrainingEvent' object（当删除活动信息时触发那些操作）。1 是删除活动信息时保留报名人员，2 是删除活动信息时删除报名人员，3 是当报名人员没有数据时可以删除。按照需求我们选择 3。

The screenshot shows the Mendix Studio Pro 8.12.4 interface. The Domain Model is open, displaying a diagram with entities: Courses, Teacher, Location, TrainingEvent, and Registration_TrainingEvent. The TrainingEvent entity is connected to the Registration_TrainingEvent association. The Properties of Association 'MyFirstModule.Registration_TrainingEvent' dialog is open, showing the 'On delete of 'TrainingEvent' object' section. The 'Delete 'Registration' object(s) as well' option is selected, which is highlighted with a red circle and the number 3. The 'Delete 'TrainingEvent' object only if it is not associated with 'Registration' object(s)' option is also highlighted with a red circle and the number 3. The 'Delete 'Registration' object(s) as well' option is highlighted with a red circle and the number 2. The 'Delete 'TrainingEvent' object only if it is not associated with 'Registration' object(s)' option is highlighted with a red circle and the number 1.

2. 选择好以后，他会让我们输入提示信息。我们输入好以后点击 Ok 按钮。就可以了。

Properties of Association 'MyFirstModule.Registration_TrainingEvent'

Common

Name: Registration_TrainingEvent

Documentation:

Multiplicity

☐ [1 - 1] One 'TrainingEvent' object is associated with one 'Registration' object

☒ [1 - *] One 'TrainingEvent' object is associated with multiple 'Registration' objects

☐ [* - *] Multiple 'TrainingEvent' objects are associated with multiple 'Registration' objects

On delete of 'TrainingEvent' object

☐ Keep 'Registration' object(s)

☐ Delete 'Registration' object(s) as well

☒ Delete 'TrainingEvent' object only if it is not associated with 'Registration' object(s)

Error message if 'TrainingEvent' object cannot be deleted

该活动已拥有报名人员，不可删除!

On delete of 'Registration' object

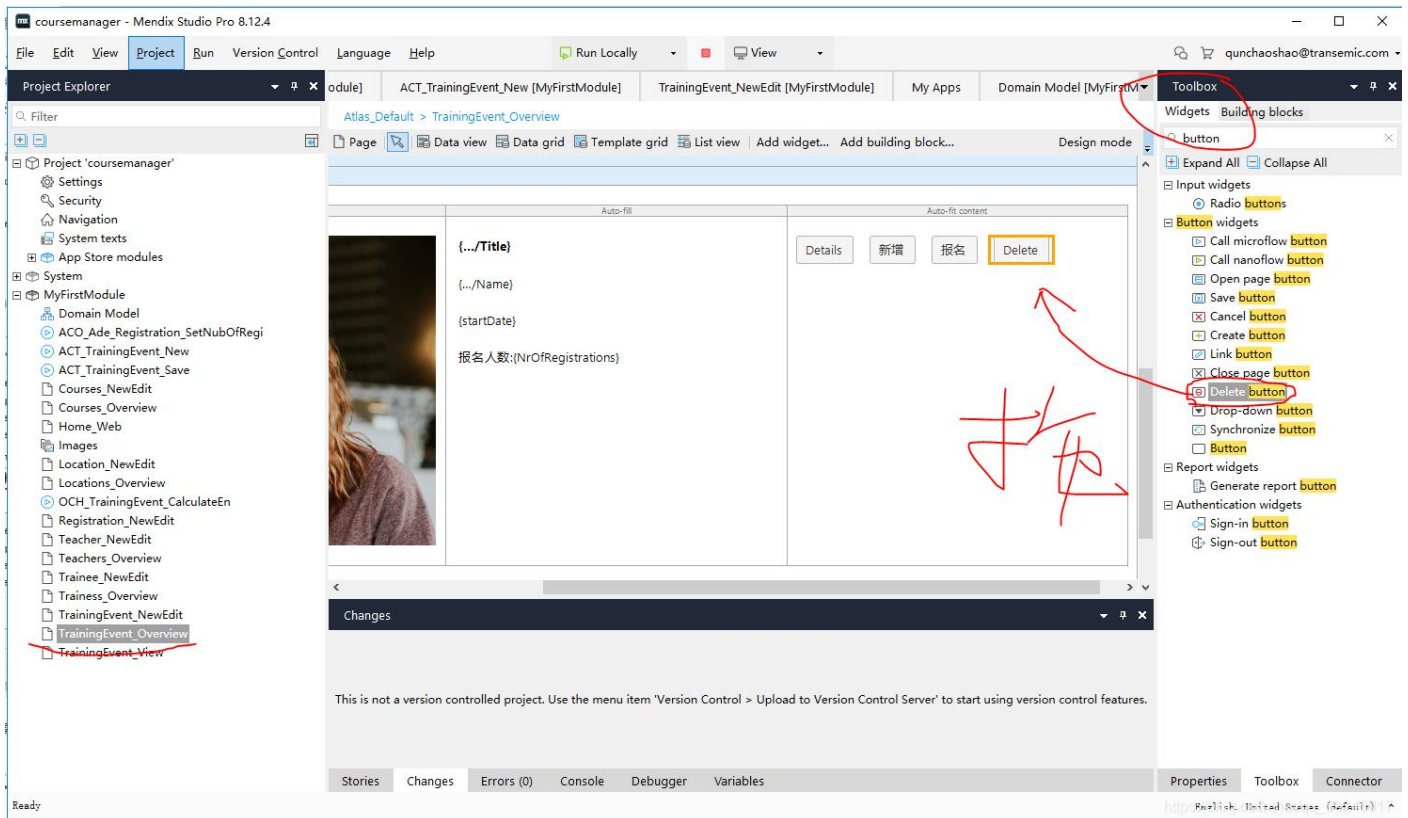
☒ Keep 'TrainingEvent' object(s)

☐ Delete 'TrainingEvent' object(s) as well

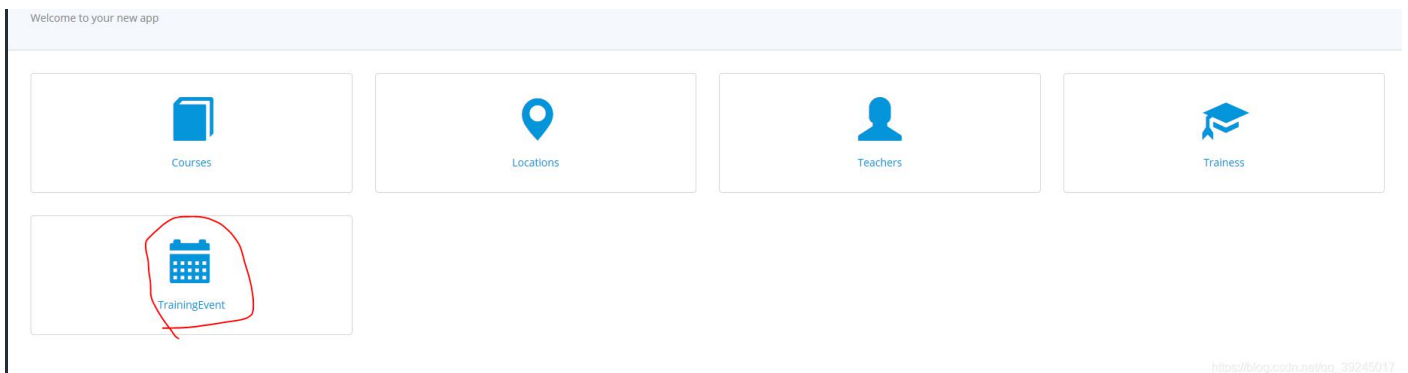
☐ Delete 'Registration' object only if it is not associated with 'TrainingEvent' object(s)

OK Cancel

3. 我们需要在活动信息列表追加删除按钮，我们在目录树上双击 **TrainingEvent_Overview** 活动列表，然后在下图圆圈区域搜索 **button**，然后找到 **Delete button** 组件拖入到页面。



4. 我们保存项目并运行测试，看效果。可以看到该功能已经实现了。



<https://blog.csdn.net/qun30245017>



JAVA高级教程

邵群超

1/14/2021

报名人数:3

Details

新增

报名

Delete

Error

该活动已拥有报名人员，不可删除！

OK

https://blog.csdn.net/qq_39245017

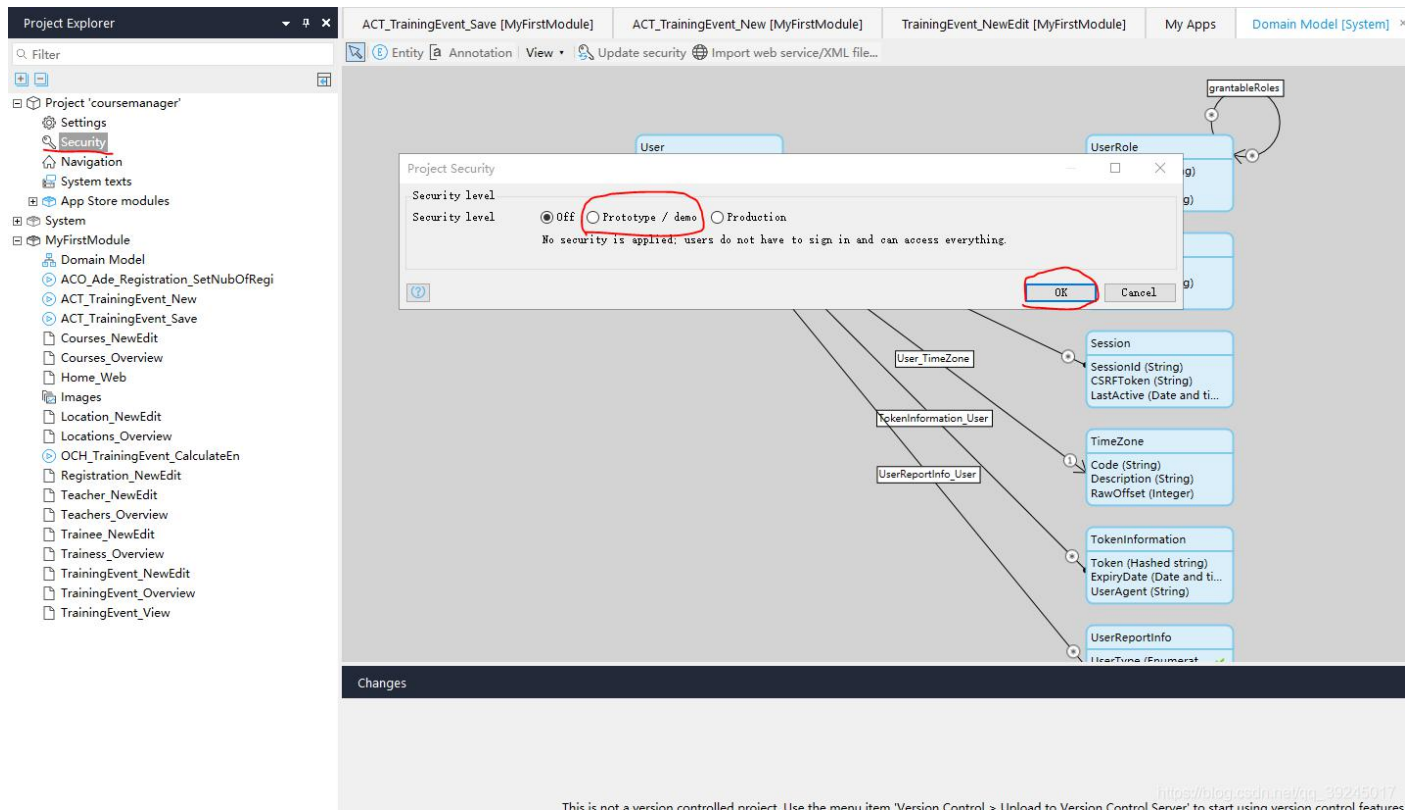
二、Mendix 权限(Security)

1、 开启系统权限 ProtoType/demo

双击左侧目录树，点击 Project Security 模块。我们选择 ProtoType/demo(测试环境)开启系统的权限。

当系统权限开启以后，每个模块下都会出现 Security 模块，Security 模块控制每个用户具体可以读写

哪些数据和访问哪些页面、以及微流！



2、项目权限(Project Security)

Project Security(项目权限)打个比方，像代表一个小区，可以管理那些用户访问具体的哪栋楼，哪个房间。

2.1 菜单功能介绍

2.1.1 Module status

该功能控制系统所有模块中的功能(Page access 页面、Nanoflow access 原生模板移动端、Microflow access 微流、Entity access 数据模型，Data set access 数据等)是否开启权限控制。

Project Security

Security level

Security level

☐ Off
☒ Prototype / demo
☐ Production

Security is applied to signing in, forms, and microflows. Configure administrator and anonymous access and define user roles and security for forms and microflows.

Project status

☒ Incomplete

Module status

User roles

Administrator

Demo users

Anonymous users

Password policy

Edit module security

Module	Page access	Nanoflow access	Microflow access	OData access	REST access	Entity access	Data set access
MyFirstModule	Incomplete	Complete	Incomplete	Complete	Complete	Complete	Complete

OK

Cancel

2.1.2 User roles

所有的角色，可以细分，新建的时候可以控制管理哪些模块。就是设置用户可以访问哪些楼。

Project Security

Security level

Security level

☐ Off
☒ Prototype / demo
☐ Production

Security is applied to signing in, forms, and microflows. Configure administrator and anonymous access and define user roles and security for forms and microflows.

Project status

☒ Incomplete

Module status

User roles

Administrator

Demo users

Anonymous users

Password policy

New

Edit

Delete

Name	Module roles
Administrator	System.Administrator, Administration.Administrator, MyFirstModule.User
User	System.User, MyFirstModule.User, Administration.User

?

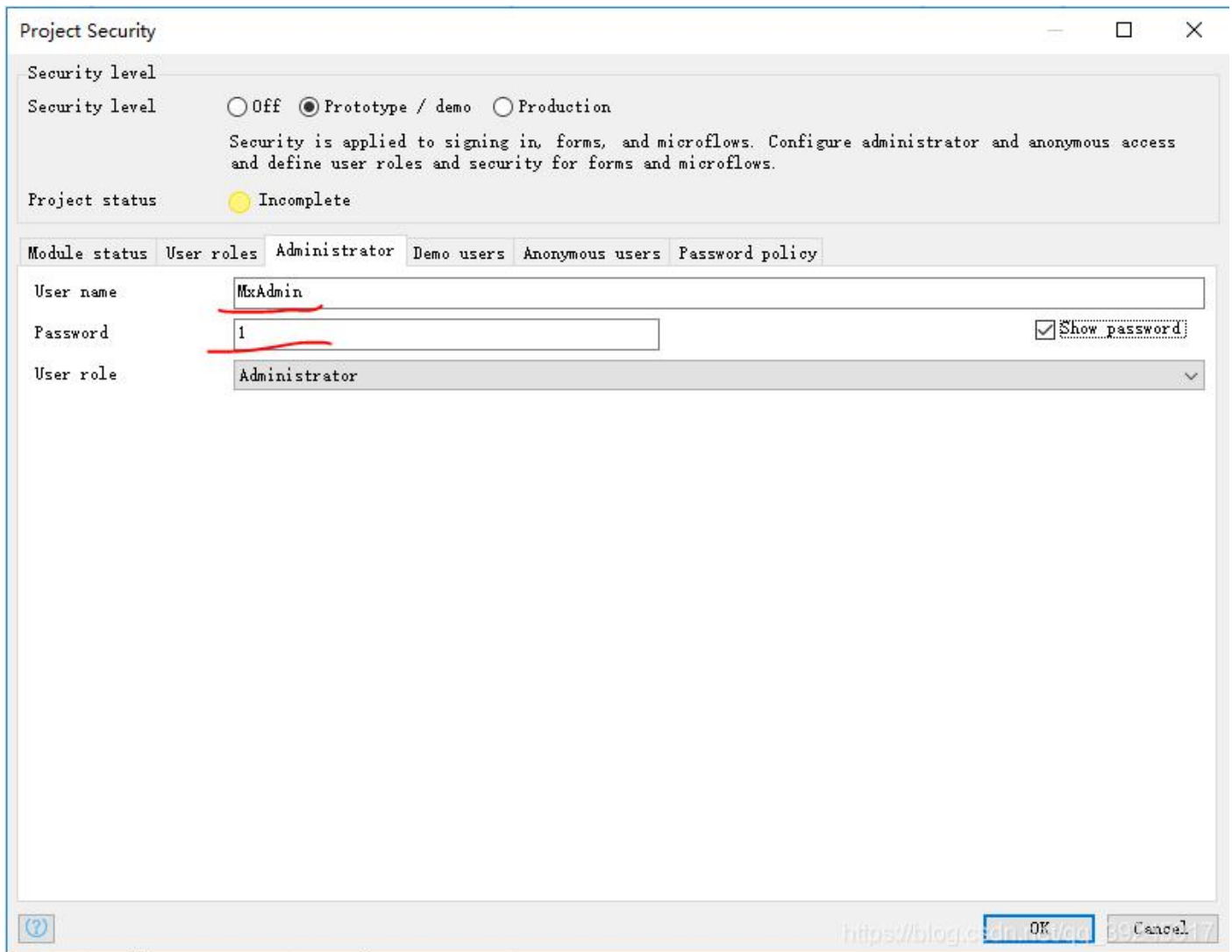
https://blog.csdn.net/qq_34117

OK

Cancel

2.1.3 Administrator

该菜单是设置超级管理员的用户名和密码。这里我默认设置用户名 MxAdmin 密码 1。在项目部署到生产环境的时候，密码需要满足复杂度的要求，详见 Password policy 介绍



Project Security

Security level
☐ Off ☒ Prototype / demo ☐ Production
 Security is applied to signing in, forms, and microflows. Configure administrator and anonymous access and define user roles and security for forms and microflows.

Project status
☒ Incomplete

Module status | **User roles** | Administrator | Demo users | Anonymous users | Password policy

User name
 MxAdmin

Password
 1 ☒ Show password

User role
 Administrator

OK Cancel

2.1.4 Demo users

Demo users 是测试环境下有那些用户，这里可以新增用户分配角色，我们可以在下边测试的时候用到。

Project Security

Security level
☐ Off ☒ Prototype / demo ☐ Production
 Security is applied to signing in, forms, and microflows. Configure administrator and anonymous access and define user roles and security for forms and microflows.

Project status ● Complete

Module status | User roles | Administrator | **Demo users** | Anonymous users | Password policy

Enable demo users ☒ Yes ☐ No
 Demo users are only available when running locally or in a Free App.

[New](#) [Edit](#) [Delete](#)

User name	Entity	User roles
demo_administrator	Account	Administrator
demo_user	Account	Teacher

[?](#) <https://blog.csdn.net/vq...> [OK](#) [Cancel](#)

2.1.5 Anonymous users

2.1.6 Password policy

该功能是密码设置规范。

Project Security

Security level

Security level

☐ Off
 ☒ Prototype / demo
 ☐ Production

Security is applied to signing in, forms, and microflows. Configure administrator and anonymous access and define user roles and security for forms and microflows.

Project status

☒ Incomplete

Module status

User roles

Administrator

Demo users

Anonymous users

Password policy

Note: If you change the password policy, then all demo user passwords are regenerated.

Minimum length

Require digit

☒ Yes
 ☐ No

Require mixed case

☒ Yes
 ☐ No

This requires the password to contain both lowercase and uppercase characters.

Require symbol

☐ Yes
 ☒ No

This requires the password to contain non-alphanumeric characters. For example: ! # \$ or %

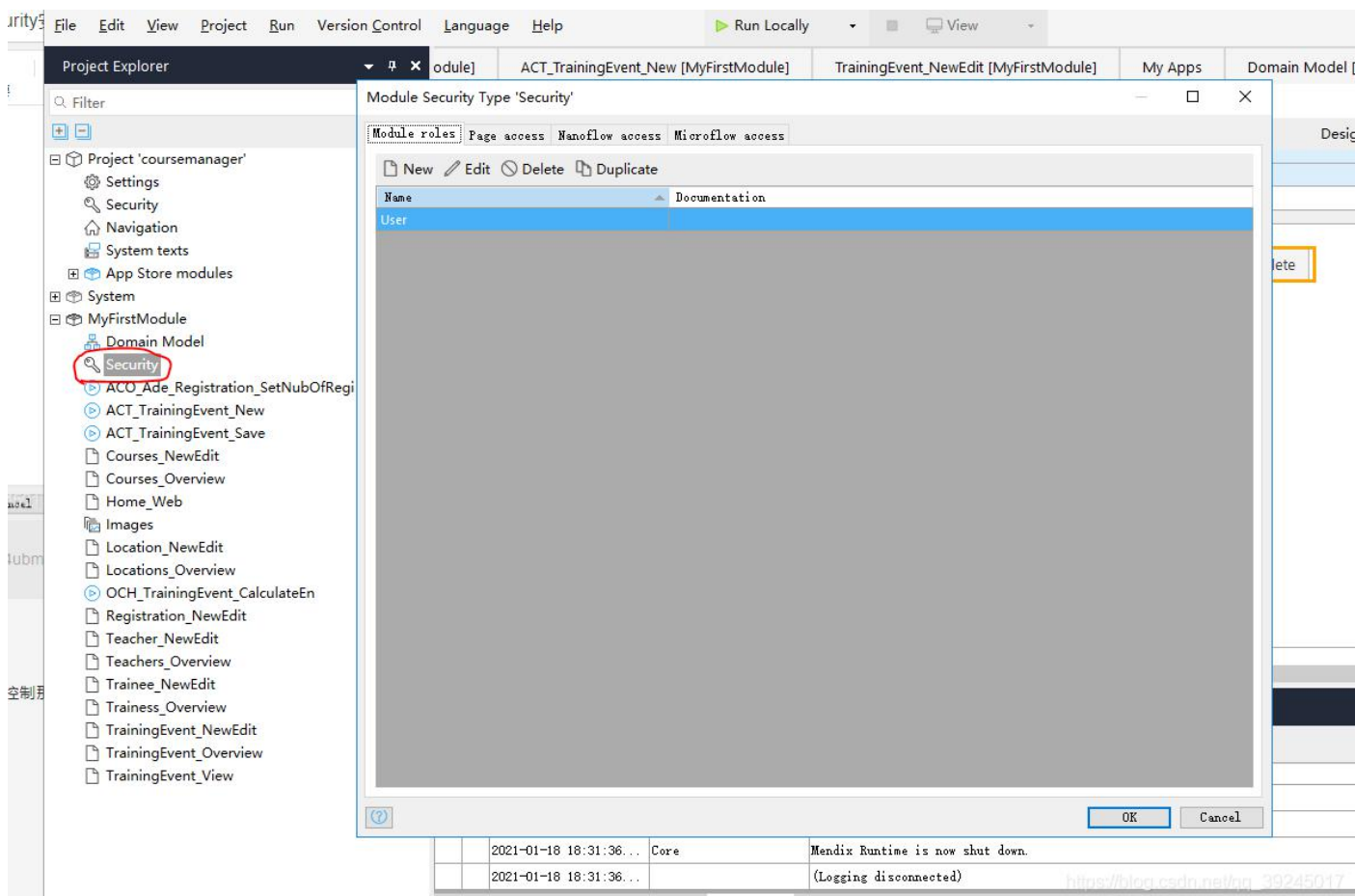
<https://blog.cdnmz.com/qq-39345177>

OK

Cancel

3、 模块权限(Module Security)

当 Project Security(系统权限)开启以后，每一个模块下会有一个 security(Module Security 模块权限)，该权限是控制那些角色可以访问模块中的不同资源。

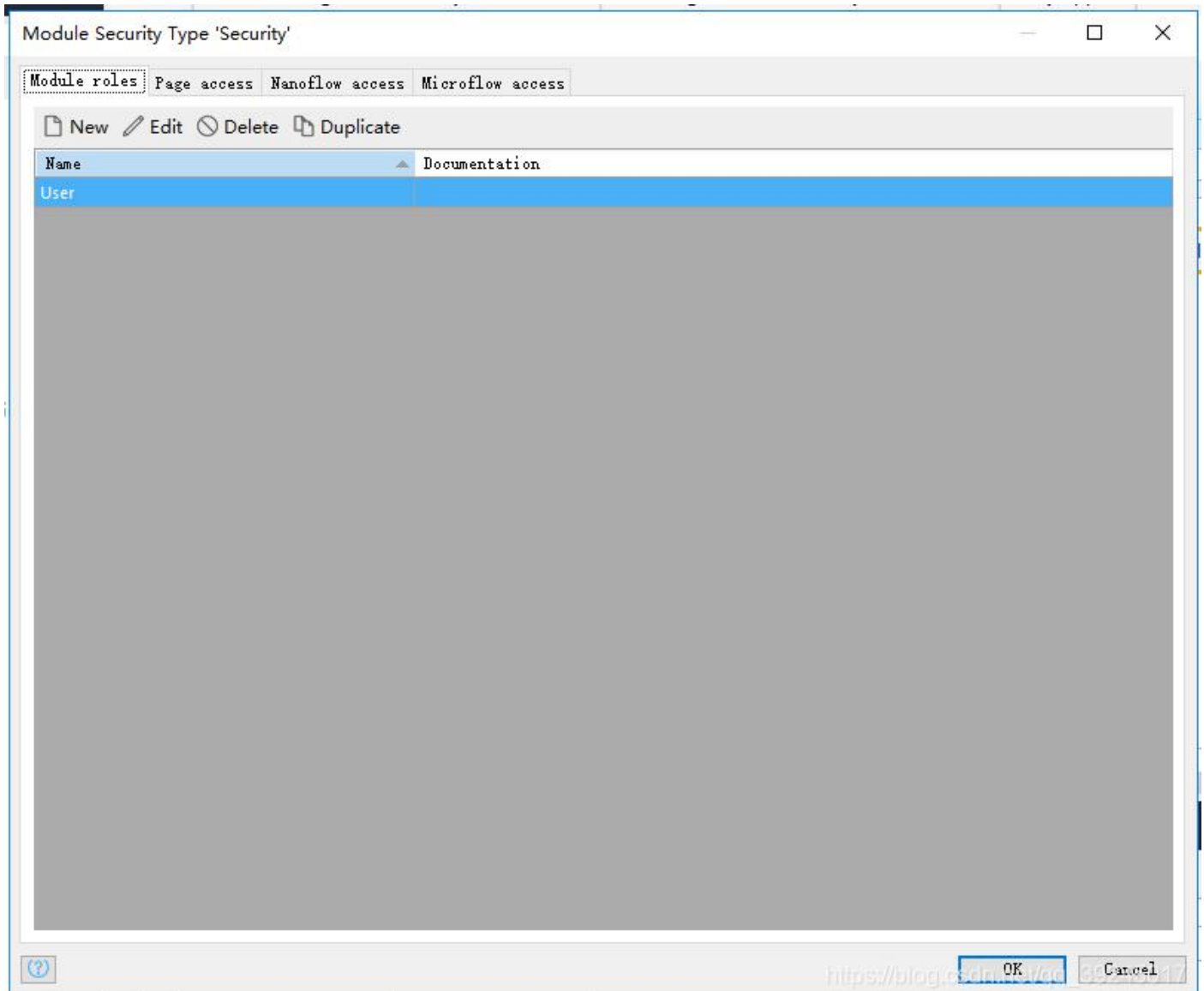


3.1 菜单功能介绍

介绍一些常用的模块。其余不常用的这里就不介绍了。以后有用到的话可以翻阅官方文档。

3.1.1 Module roles

模块角色，是展示当前模块下可以访问的所有角色，这里可以新增，编辑删除。



3.1.2 Page access

这里是设置那些角色可以访问那些页面。

Module Security Type 'Security'

Module roles
Page access
Nanoflow access
Microflow access

Note: Changes you make here modify the corresponding pages.

Open page

Page	User	Type
Courses_NewEdit	☐	Page
Courses_Overview	☐	Page
Home_Web	☒	Page
Location_NewEdit	☐	Page
Locations_Overview	☐	Page
Registration_NewEdit	☐	Page
Teacher_NewEdit	☐	Page
Teachers_Overview	☐	Page
Trainee_NewEdit	☐	Page
Trainess_Overview	☐	Page
TrainingEvent_NewEdit	☐	Page
TrainingEvent_Overview	☐	Page
TrainingEvent_View	☐	Page

Show usage

Document	Object
Page 'MyFirstModule.Courses_Overview'	Action button 'createButton2'

?

https://blog.csdn.net/qq_35555177

OK

Cancel

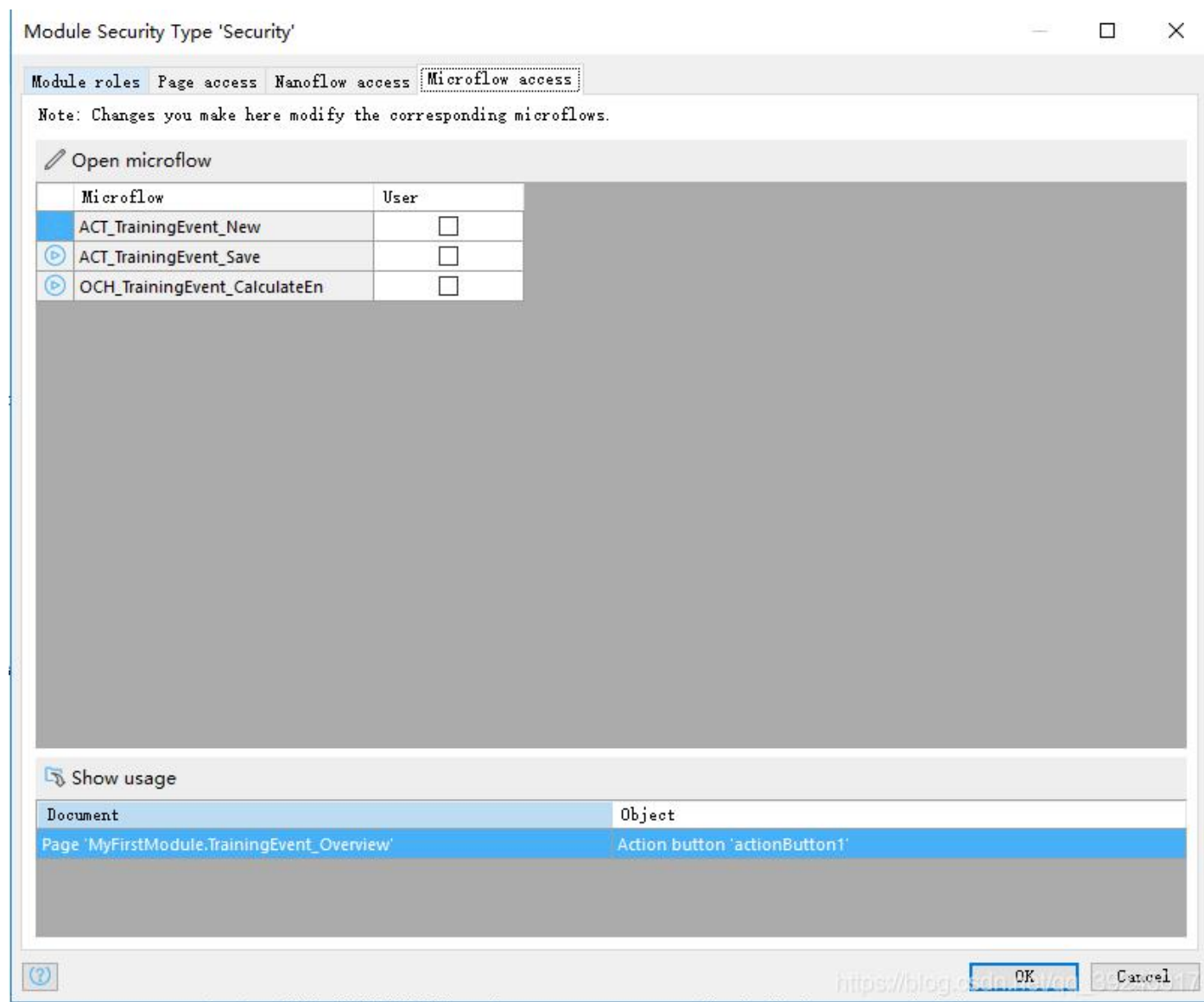
2021-11-18 18:31:36

More

Mendix Runtime is now shut down

3.1.3 Microflow access

这里设置那些角色可以访问那些 Microflow 微流。



4、项目配置

通过以上两个章节，我们大概了解了 Mendix 对 Security 权限有那些功能，现在我们实战操作一下。

4.1 项目需求

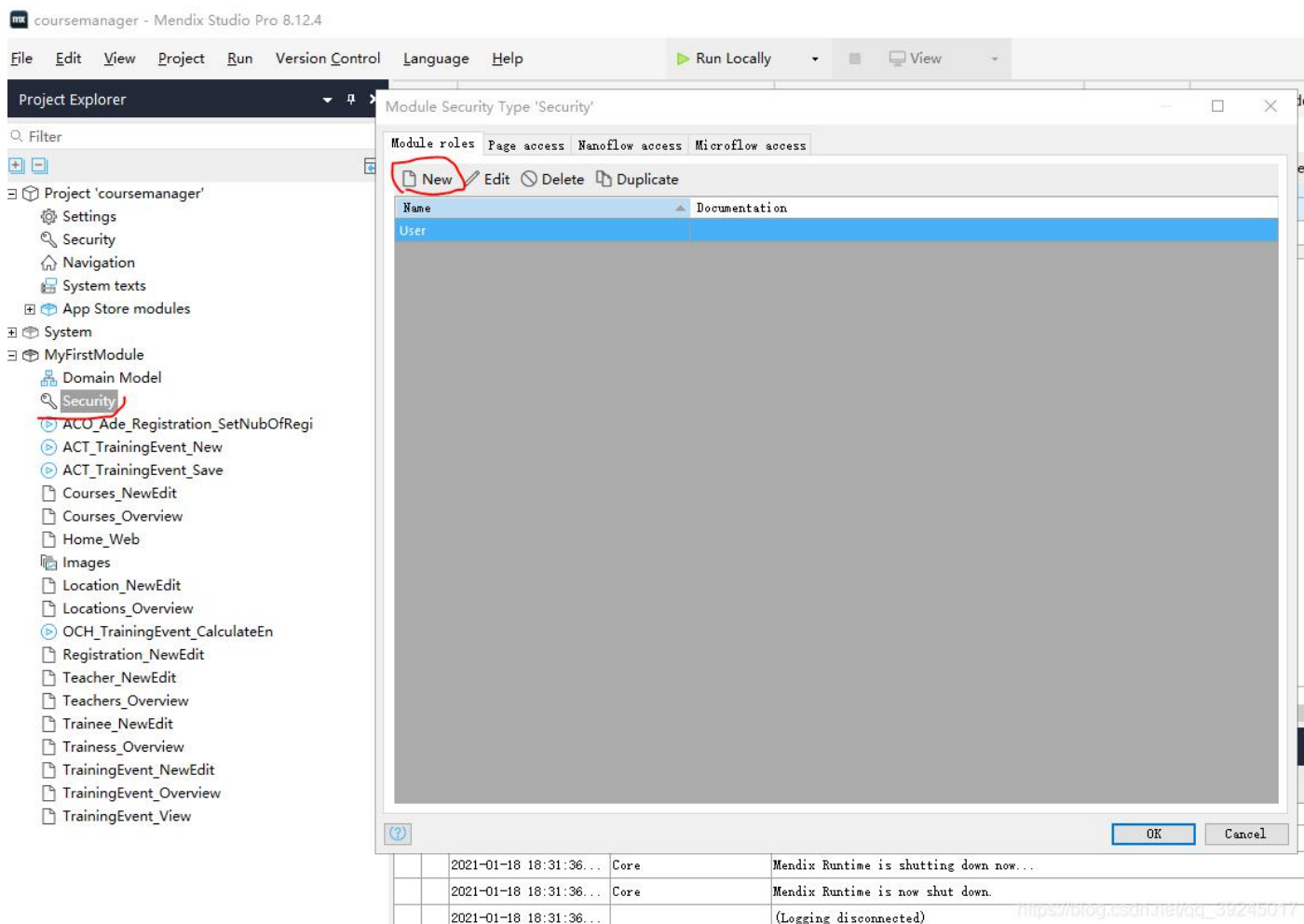
现在学校想着让老师、报名人员、和管理员访问不同的页面。实现权限划分。

4.2 需求分析

需求明确了，那我们首先需要创建三个角色。

4.3 项目实施

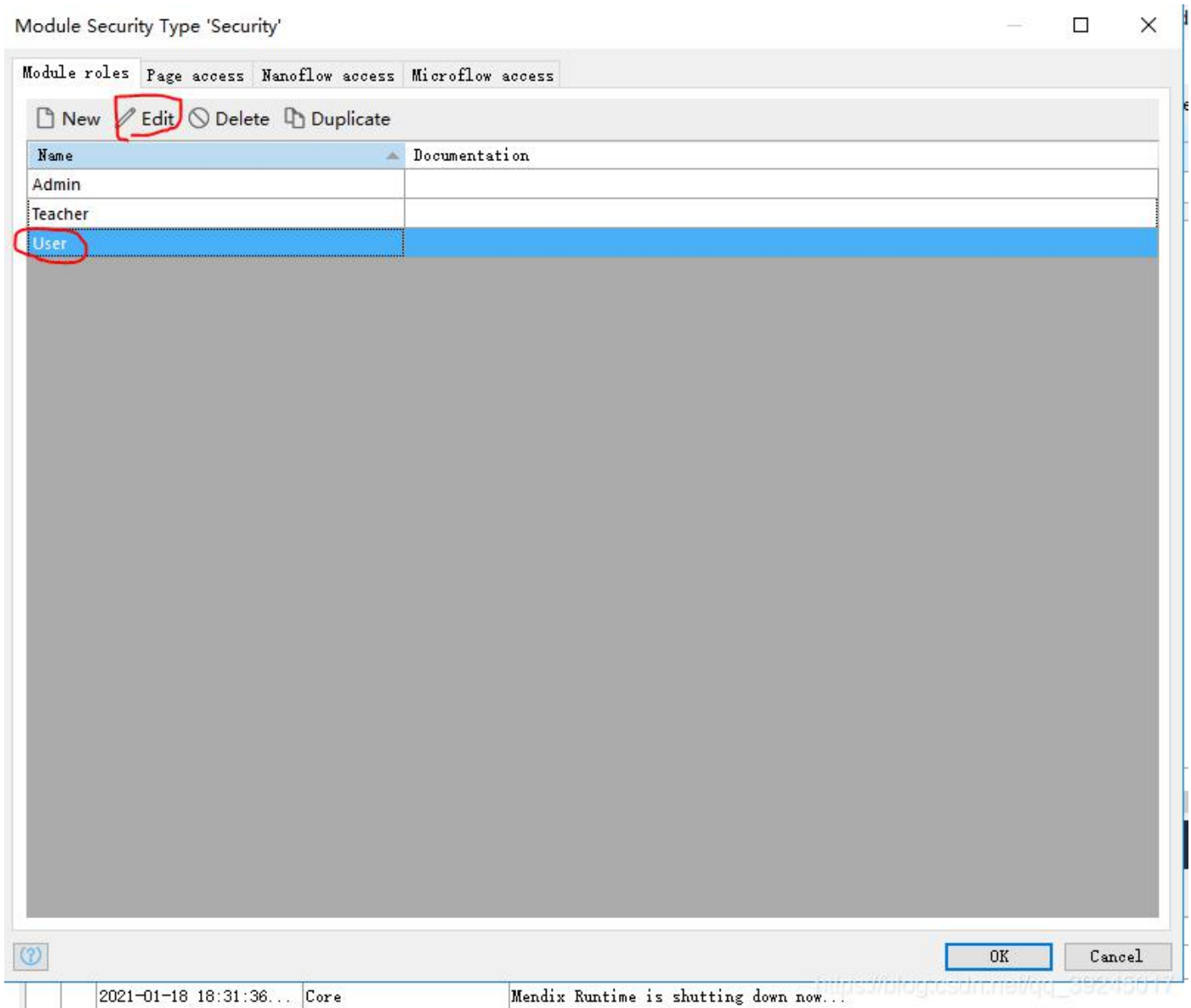
1. 我们开始实现，首先我们点开模块权限(Module Security)，然后点击 New 创建角色。



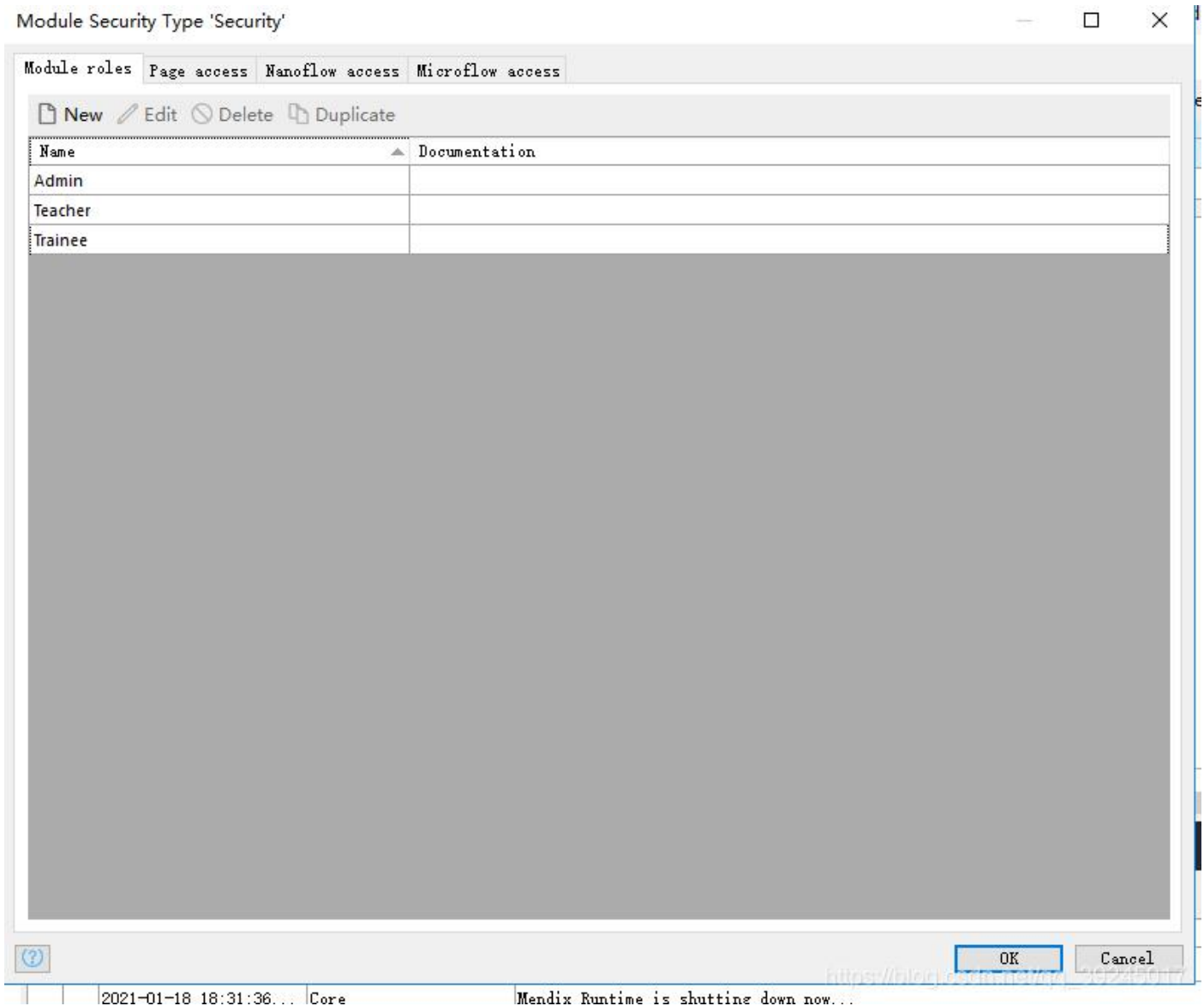
2. 在弹出的页面输入 Name 为 Admin 然后点击 Ok 按钮。我这里只展示如何创建角色，其余角色创建步骤都一样。还需要创建 Teacher(老师)和 Trainee(报名人员)。



3. 创建完 Teacher 老师角色以后,我们将 User 角色修改为 Trainee。



4. 最终创建好三个模块权限(Module Security)角色



5. 创建好模块权限(Module Security)角色以后，我们需要在项目权限(Project Security)创建同样的三种角色。双击 Project Security，然后打开 User roles，我们可以看到已经有两个角色权限，Administrator

就是我们模块里对应的 admin 角色。所以管理员角色就不用我们创建了。

Project Security

Security level: ☐ Off ☒ Prototype / demo ☐ Production

Security is applied to signing in, forms, and microflows. Configure administrator and anonymous access and define user roles and security for forms and microflows.

Project status: ● Incomplete

Module status: **User roles** Administrator Demo users Anonymous users Password policy

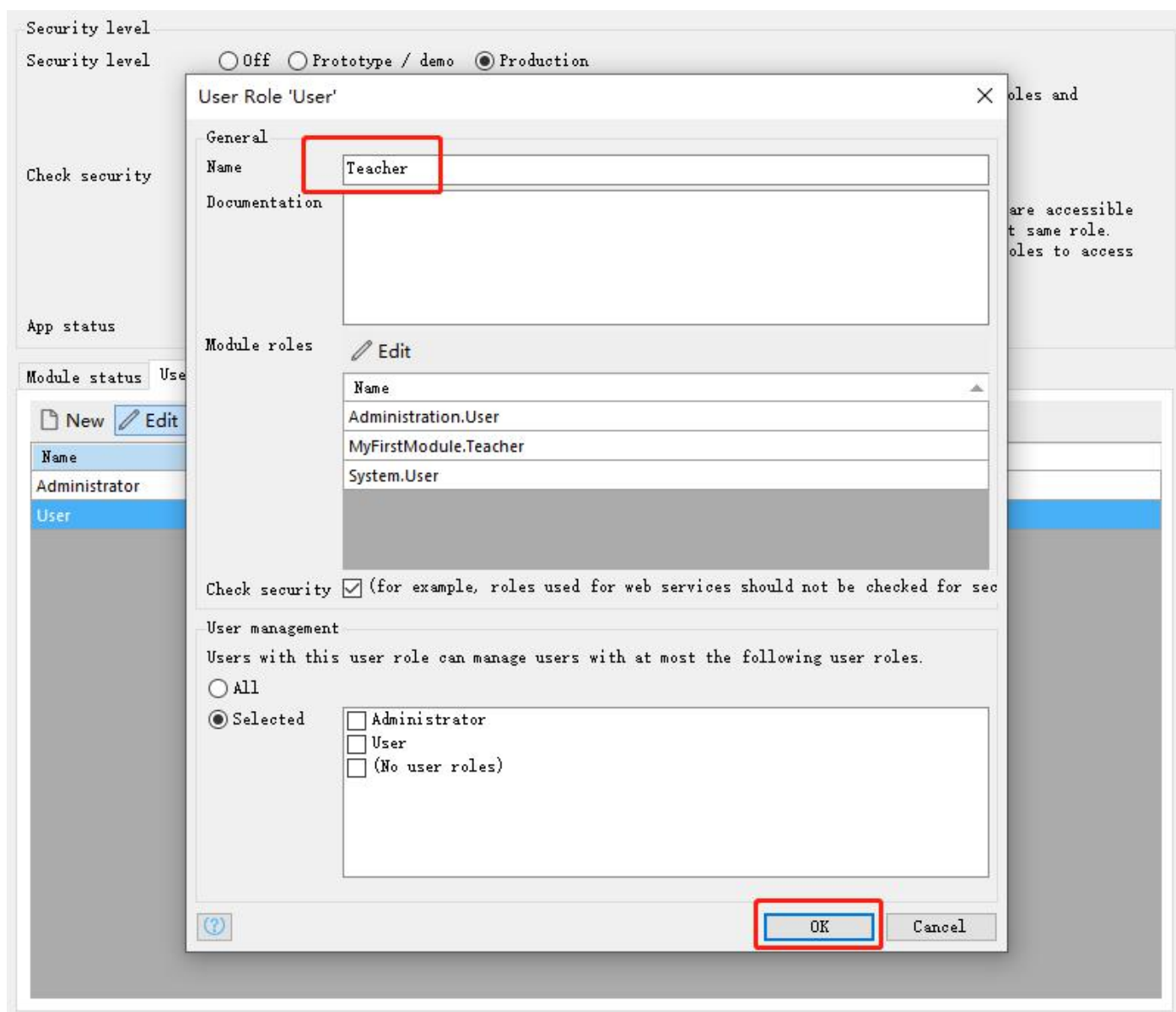
Name	Module roles
Administrator	System.Administrator, Administration.Administrator, MyFirstModule.Trainee
User	System.User, MyFirstModule.Trainee, Administration.User

Buttons: New Edit Delete

Buttons: OK Cancel

Date/time	Log node	Message
2021-01-18 18:26:44...	Core	Application model has been updated, application is now available.
2021-01-18 18:31:36...	Core	Mendix Runtime is shutting down now...

6. 然后将 User 权限修改为 Teacher 角色。选中 User 点击 Edit 按钮，在弹出的页面修改 Name 为 Teacher，然后点击 Ok 按钮。完成修改。



7. 然后我们需要再创建一个报名人员的角色 Trainee，点击 New 按钮，在弹出的页面修改 Name 为 Trainee，其中 Create module role in modules 是询问我们是否需要在模块权限中创建相同的角色，因为我们刚开始已经创建了，所以这时候就取消勾选就可以了。如果刚开始在 Project Security 中创建角色模块中就不用创建了(便利性)。然后点击 Ok 按钮就可以了。

8. **MyFirstModule** 是我们创建的模块、**Administration** 和 **Atles_***** 是 Mendix 的系统模块。

App Security

Security level

Security level ☐ Off ☐ Prototype / demo ☒ Production

Full security is applied. Configure administrator and anonymous users and security for forms, microflows, entities, and reports.

Check security ☒ Yes ☐ No

If there are no other errors, Mendix Studio Pro checks your application for a certain role only refer to attributes and associations. This assumes that each user role is independent and that it has its own functionality in your application.

App status ● Incomplete

Module status User roles Administrator Demo users Anonymous users Passwords

New Edit Delete	
Name	Module roles
Administrator	System.Administrator, Administration.Administrator, MyFirstModule.Administrator
Teacher	Administration.User, System.User, MyFirstModule.Teacher

Add User Role

Name

Trainee

Create module role in modules:

☐ MyFirstModule
 ☐ Administration
 ☐ Atlas_Core
 ☐ Atlas_Web_Content
 ☐ DataGrid

☐ Select / deselect all

OK

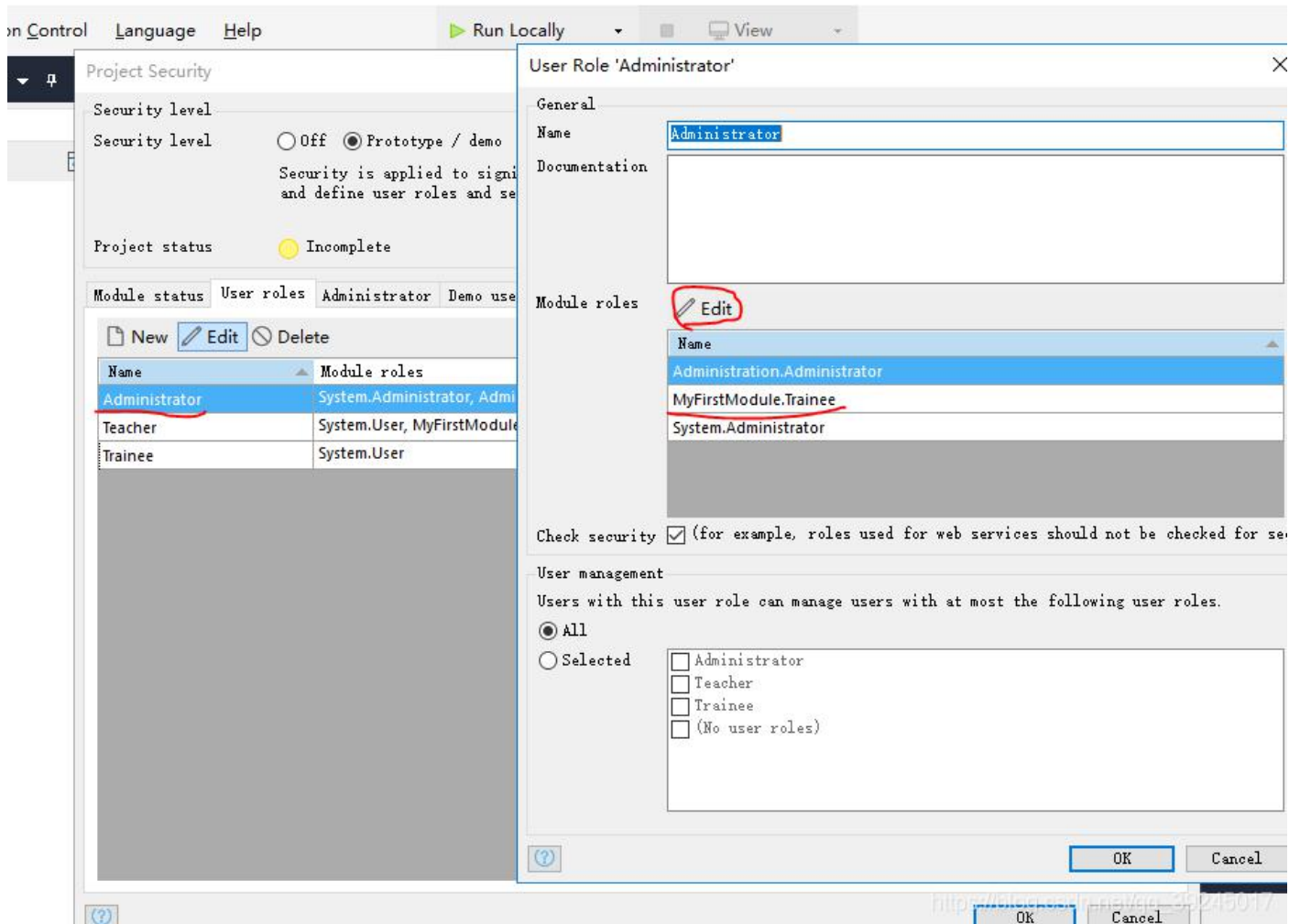
Cancel



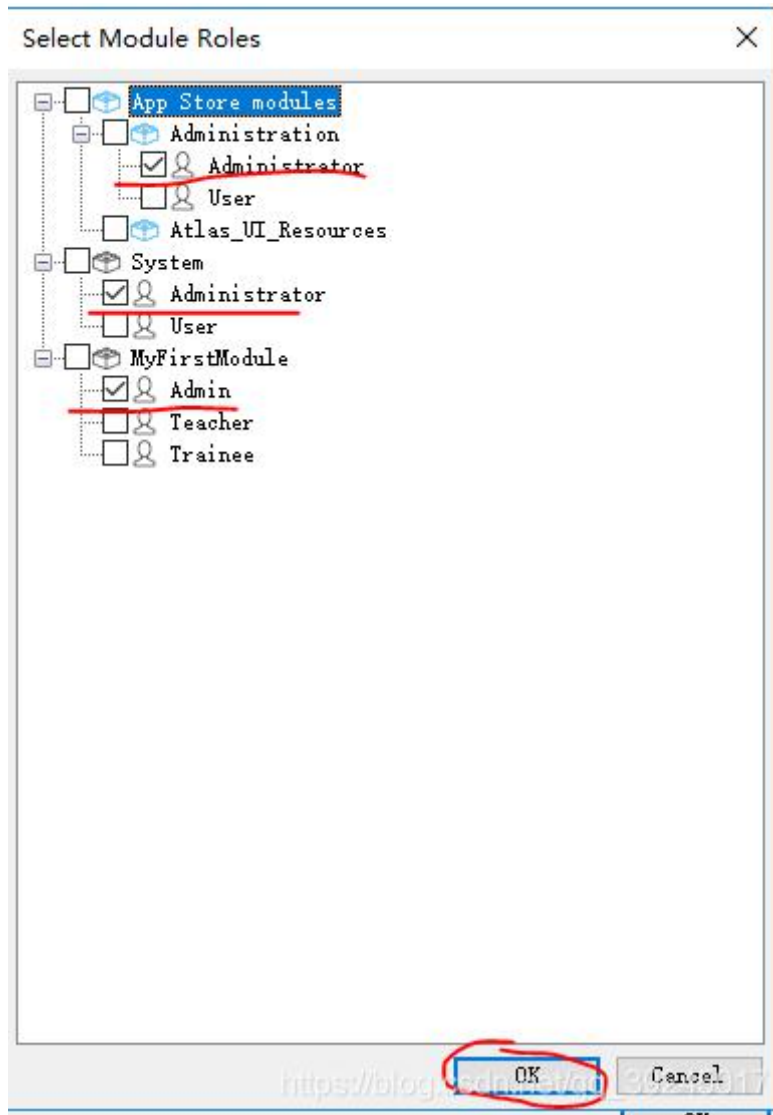
OK

Cancel

8. 创建好角色以后我们开始修改角色对应模块中的那些角色。我们先看 Administrator 管理员的角色，可以看到它对应的是我们模块 MyFirstModule 的 Trainee 所以是不对的，我们点击 Edit 按钮进行修改。



9. 我们将 Administrator 选择对应所有模块下的管理员。然后点击 Ok-Ok 按钮完成选择。



mx mendix
A Siemens Business

User Role 'Administrator'

General

Name:

Documentation:

Module roles: [Edit](#)

Name
Administration.Administrator
MyFirstModule.Admin
System.Administrator

Check security ☒ (for example, roles used for web services should not be checked for security)

User management

Users with this user role can manage users with at most the following user roles.

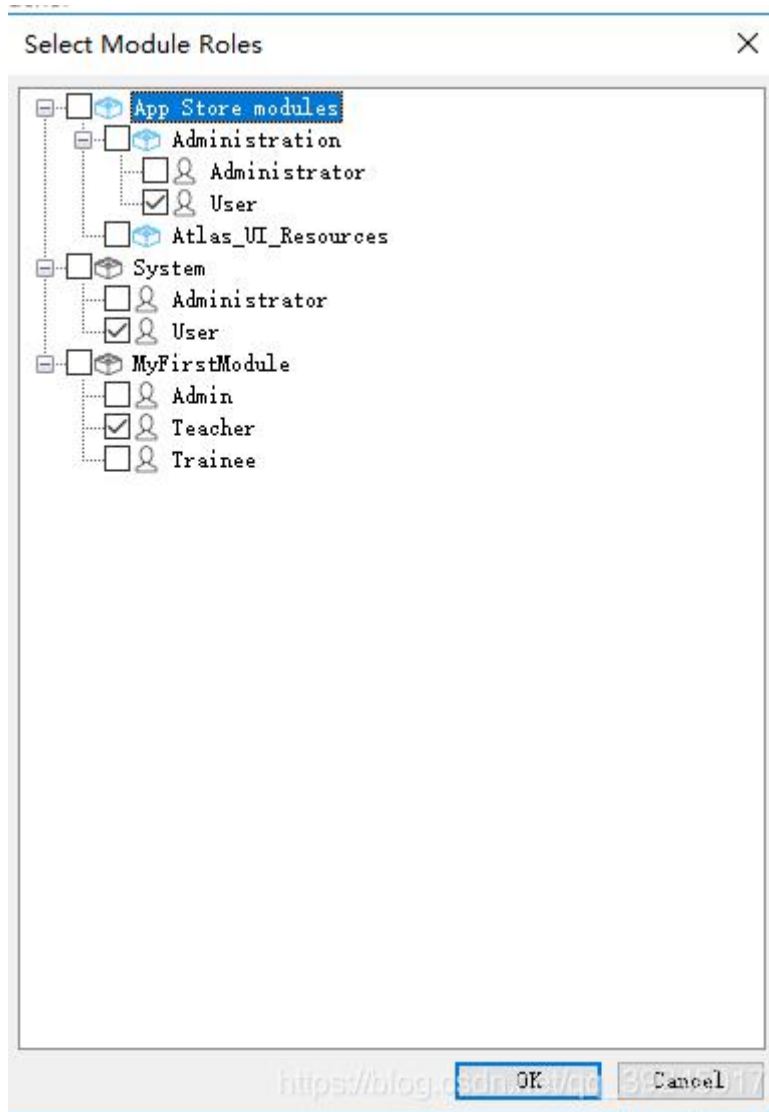
☒ All

☐ Selected

☐ Administrator
☐ Teacher
☐ Trainee
☐ (No user roles)

[?](#) <https://blog.siemens-mendix.com/39/2017/01/17/> **OK** **Cancel**

10. 接着我们检查 Teacher 角色



User Role 'Teacher'

×

General

Name

Teacher

Documentation

Module roles

✎ Edit

Name

Administration.User

MyFirstModule.Teacher

System.User

Check security

☒

(for example, roles used for web services should not be checked for security)

User management

Users with this user role can manage users with at most the following user roles.

☐ All

☒ Selected

☐ Administrator

☐ Teacher

☐ Trainee

☐ (No user roles)

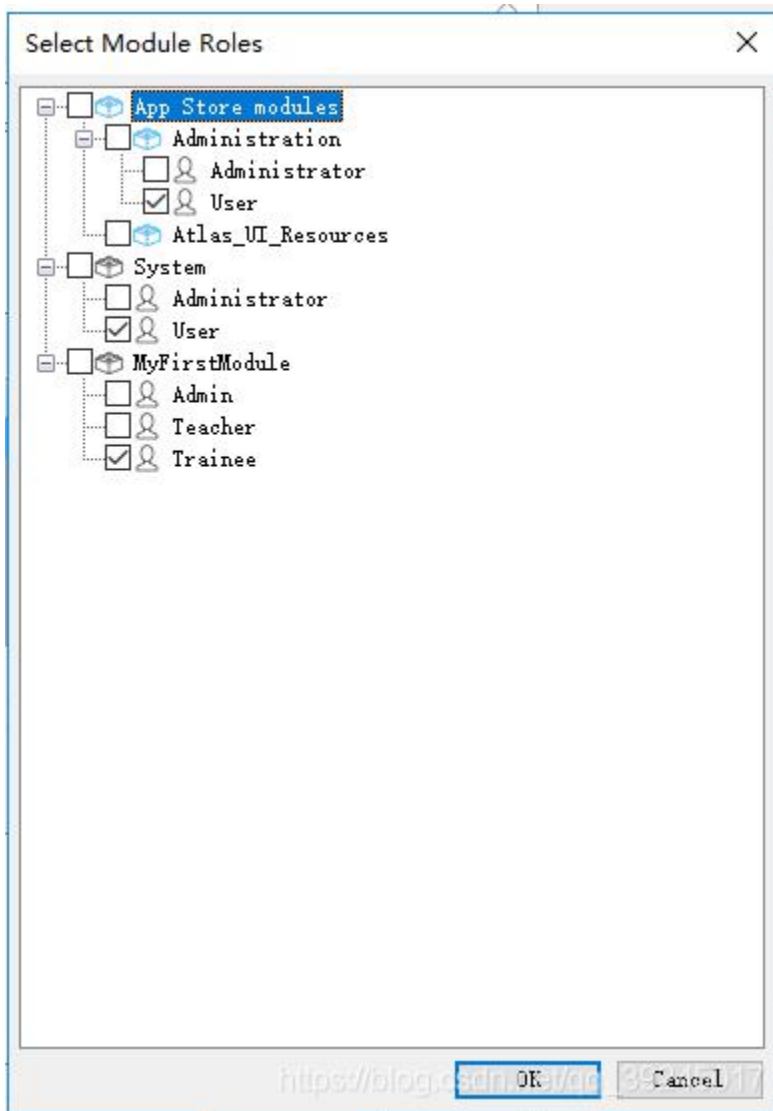
?

https://blog.sdr-OK.de/3644017

OK

Cancel

11. 检查一下 Trainee 角色



User Role 'Trainee'

General

Name

Documentation

Module roles

Name
Administration.User
MyFirstModule.Trainee
System.User

Check security ☒ (for example, roles used for web services should not be checked for security)

User management

Users with this user role can manage users with at most the following user roles.

☐ All

☒ Selected

☐ Administrator
☐ Teacher
☐ Trainee
☐ (No user roles)

<https://blog.csdn.net/397191717>

12. 到这一步，我们完成所有角色的创建了。我们开始设置角色的权限，双击模块下的 Security，然后打开 Page access 页面，按照下图进行配置。管理员是拥有所有的权限，Teacher 是可以看到所有的详

情页面 Overview 和活动页面。Trainee 是只能看到活动列表。配置好以后点击 Ok 按钮。

Module Security Type 'Security'

Module roles: **Page access** Nanoflow access Microflow access

Note: Changes you make here modify the corresponding pages

Open page

Page	Admin	Teacher	Trainee	Type
Courses_NewEdit	☒	☐	☐	Page
Courses_Overview	☒	☒	☐	Page
Home_Web	☒	☒	☐	Page
Location_NewEdit	☒	☐	☐	Page
Locations_Overview	☒	☒	☐	Page
Registration_NewEdit	☒	☐	☐	Page
Teacher_NewEdit	☒	☐	☐	Page
Teachers_Overview	☒	☒	☐	Page
Trainee_NewEdit	☒	☐	☐	Page
Trainess_Overview	☒	☒	☐	Page
TrainingEvent_NewEdit	☒	☐	☐	Page
TrainingEvent_Overview	☒	☒	☒	Page
TrainingEvent_View	☒	☒	☐	Page

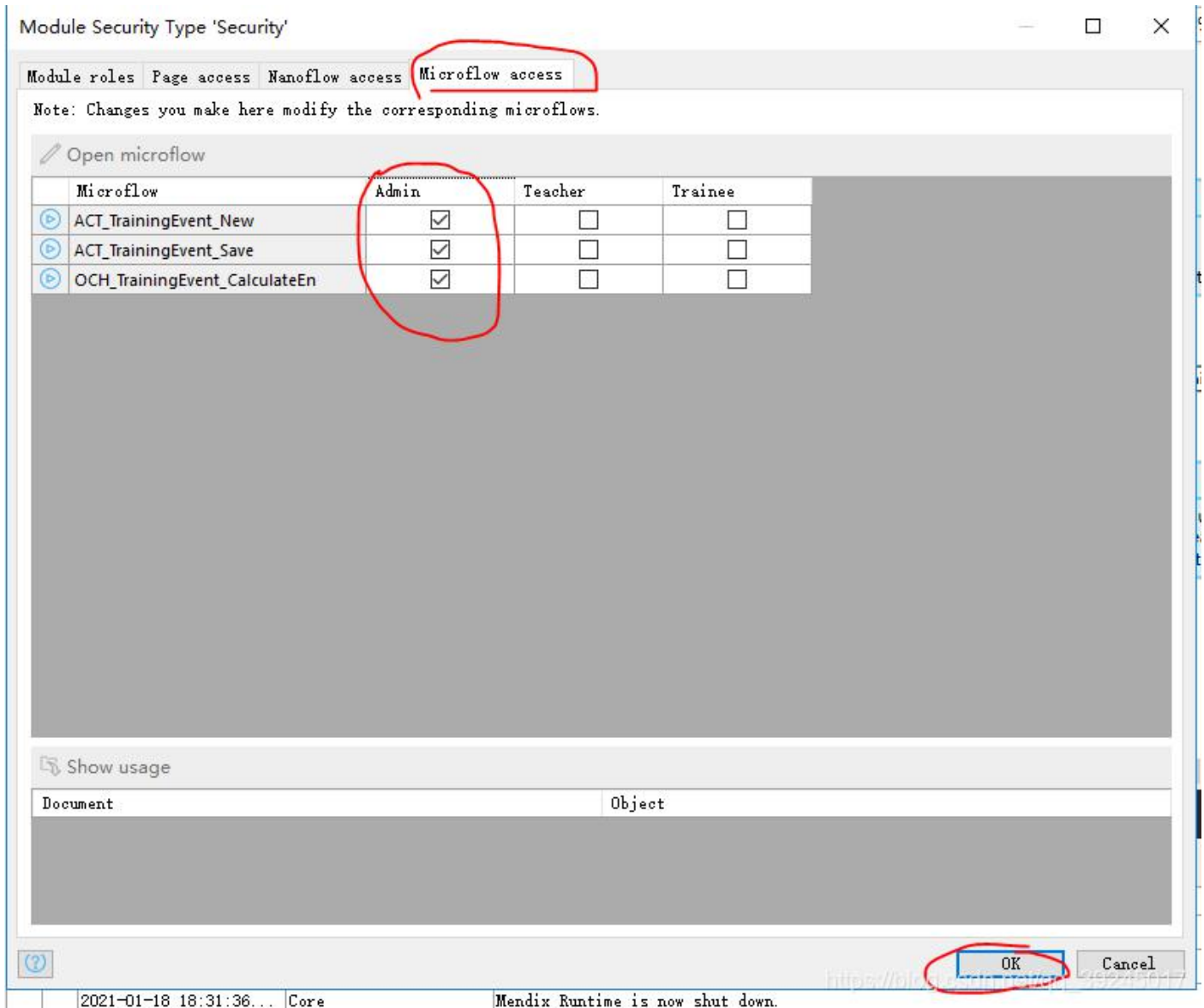
Show usage

Document	Object
Page: MyFirstModule.Courses_Overview	Action button createButton2

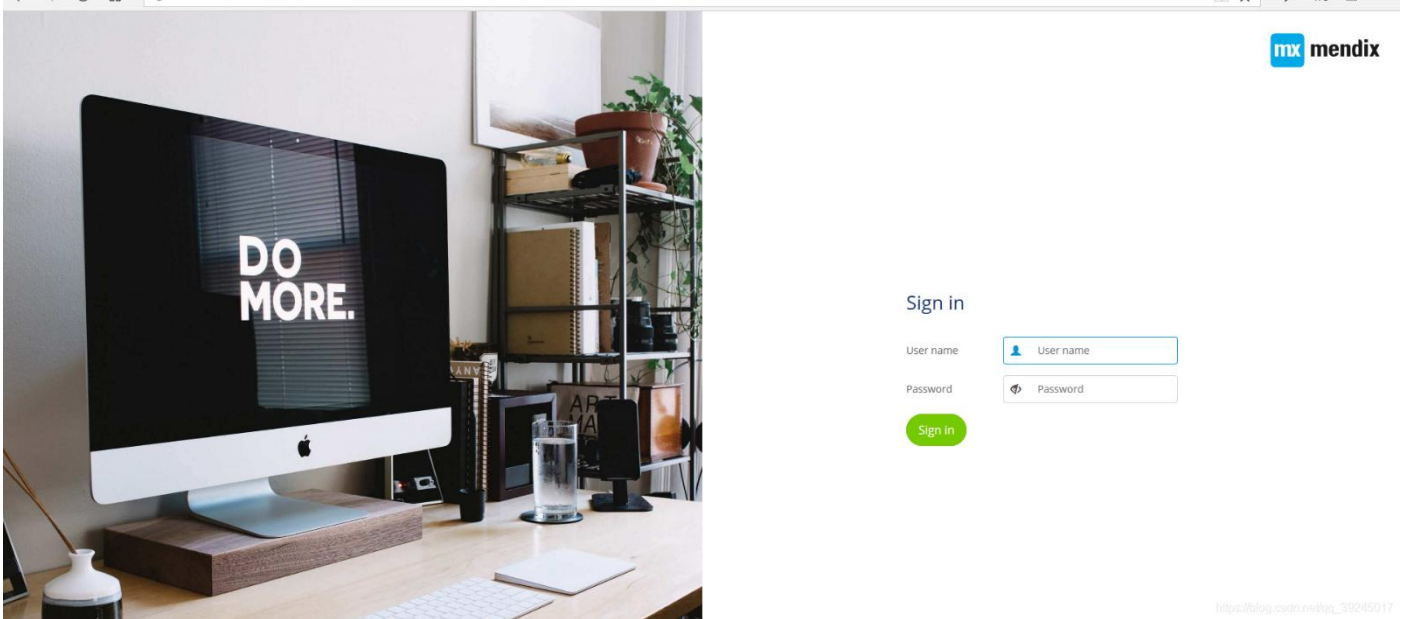
OK Cancel

2021-01-18 18:31:36... Core Mendix Runtime is now shut down.
(Logging disconnected)

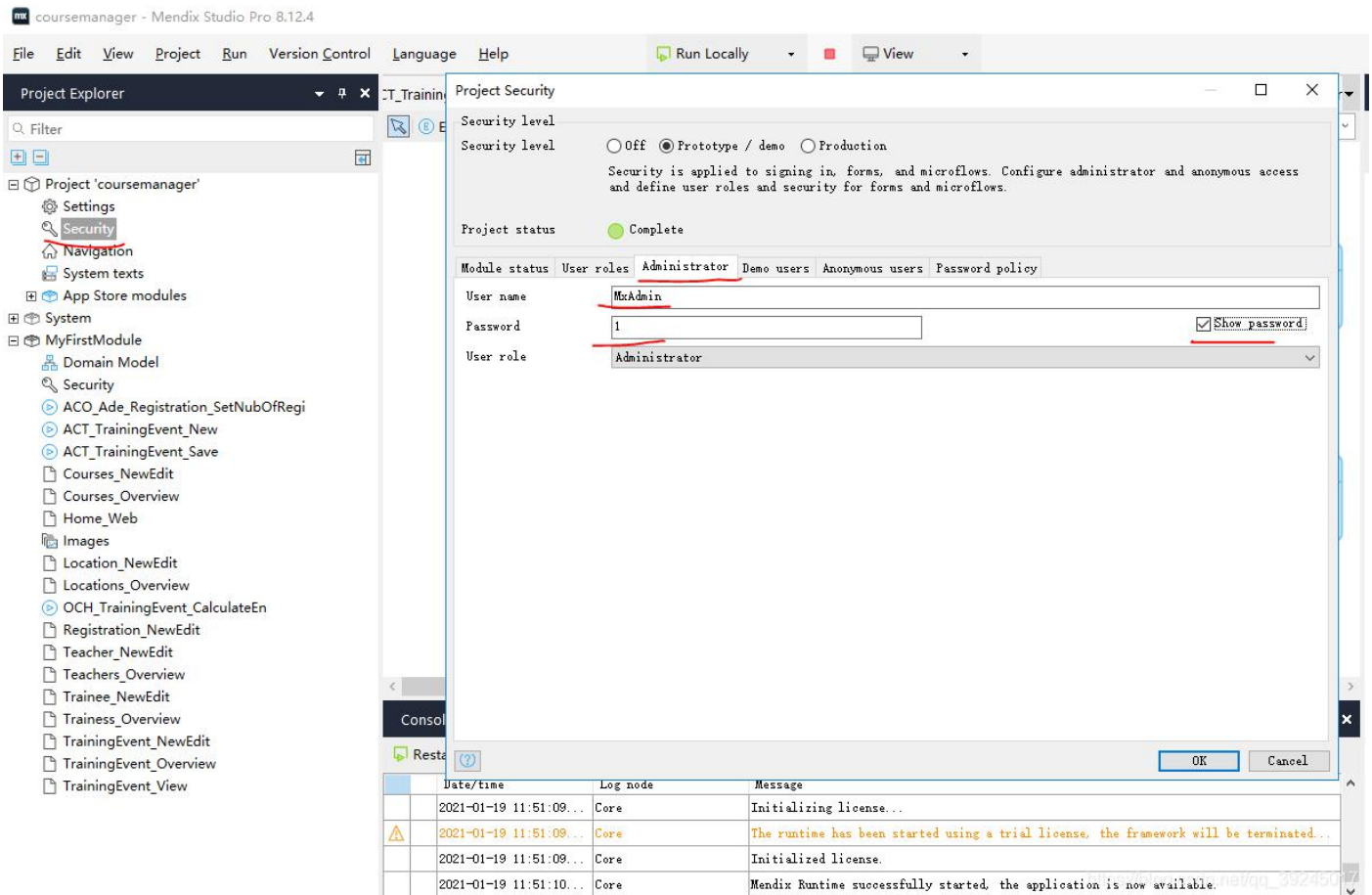
13. 我们对应的需要配置一下 Microflow access 的权限，应该目前我们创建的 Microflow 都是关于写数据的一些操作，所以我们只需要管理员拥有这个权限就可以了。然后点击 Ok 按钮就可以了。



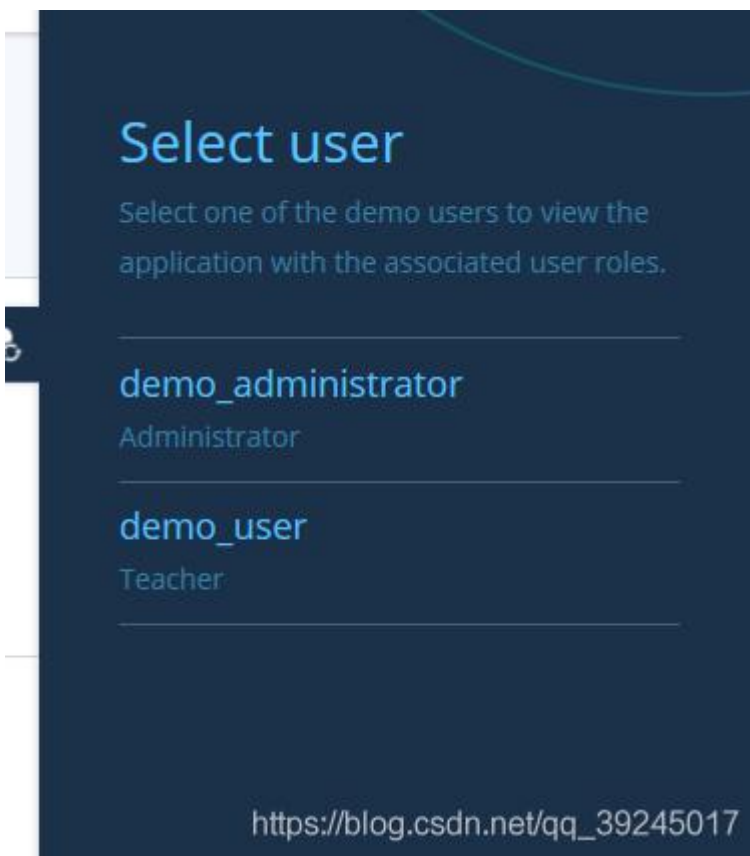
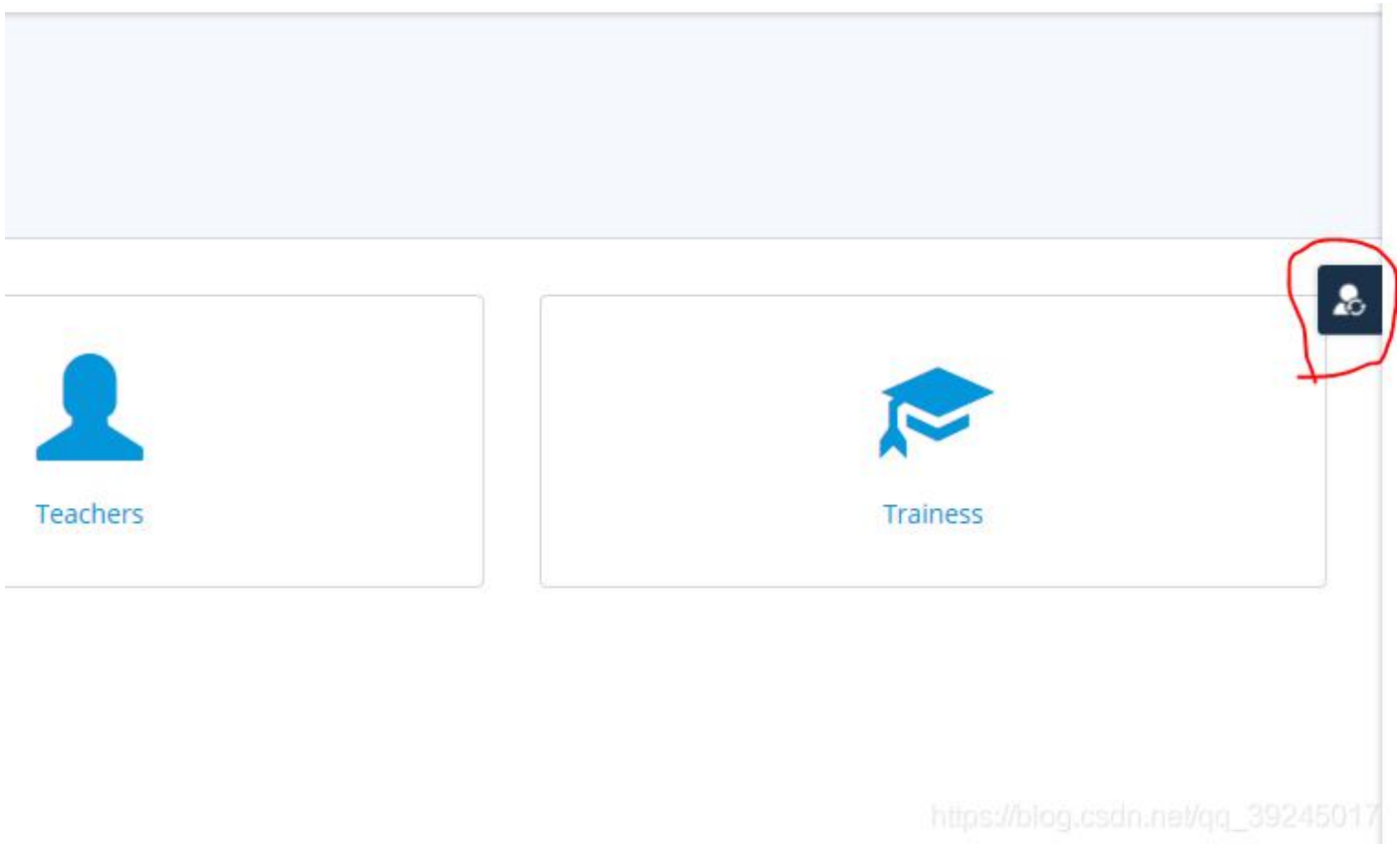
14. 配置好权限以后，我们运行项目，点击 run locally，打开浏览器进行访问，我们发现需要登陆：



15. 我们可以在 Project Security 下的 Administrator 中看到管理员的用户和密码。



16. 我们现在可以登陆一下，发现系统右侧会有一个小图标，我们点击它可以进行用户的切换。



17. 我们可以进行用户的切换，对比页面上有什么不同。我们可以发现 demo_administrator 是拥有新增按钮的，demo_user 是没有新增按钮的。

Title

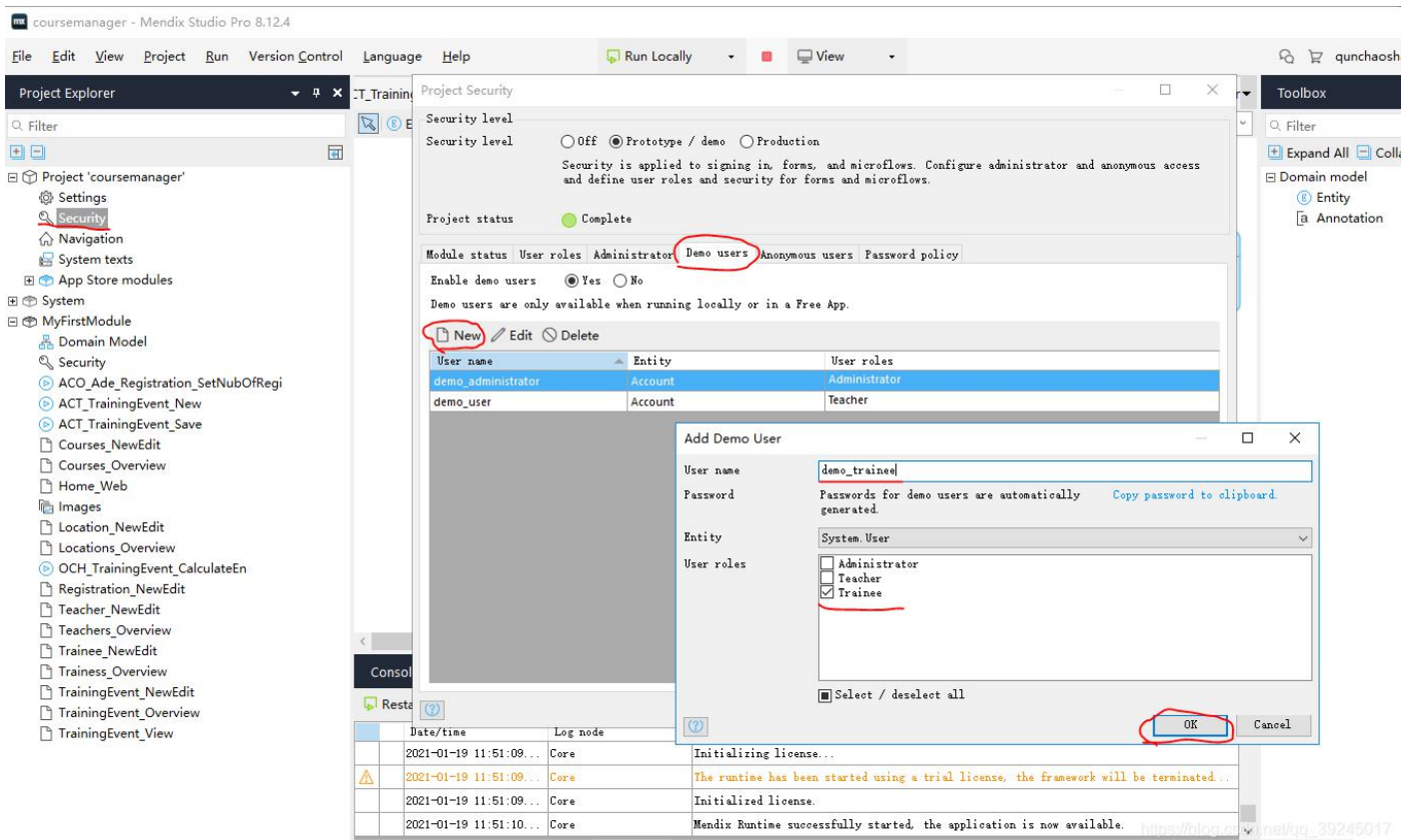
Here you can put a paragraph as subtitle

Title

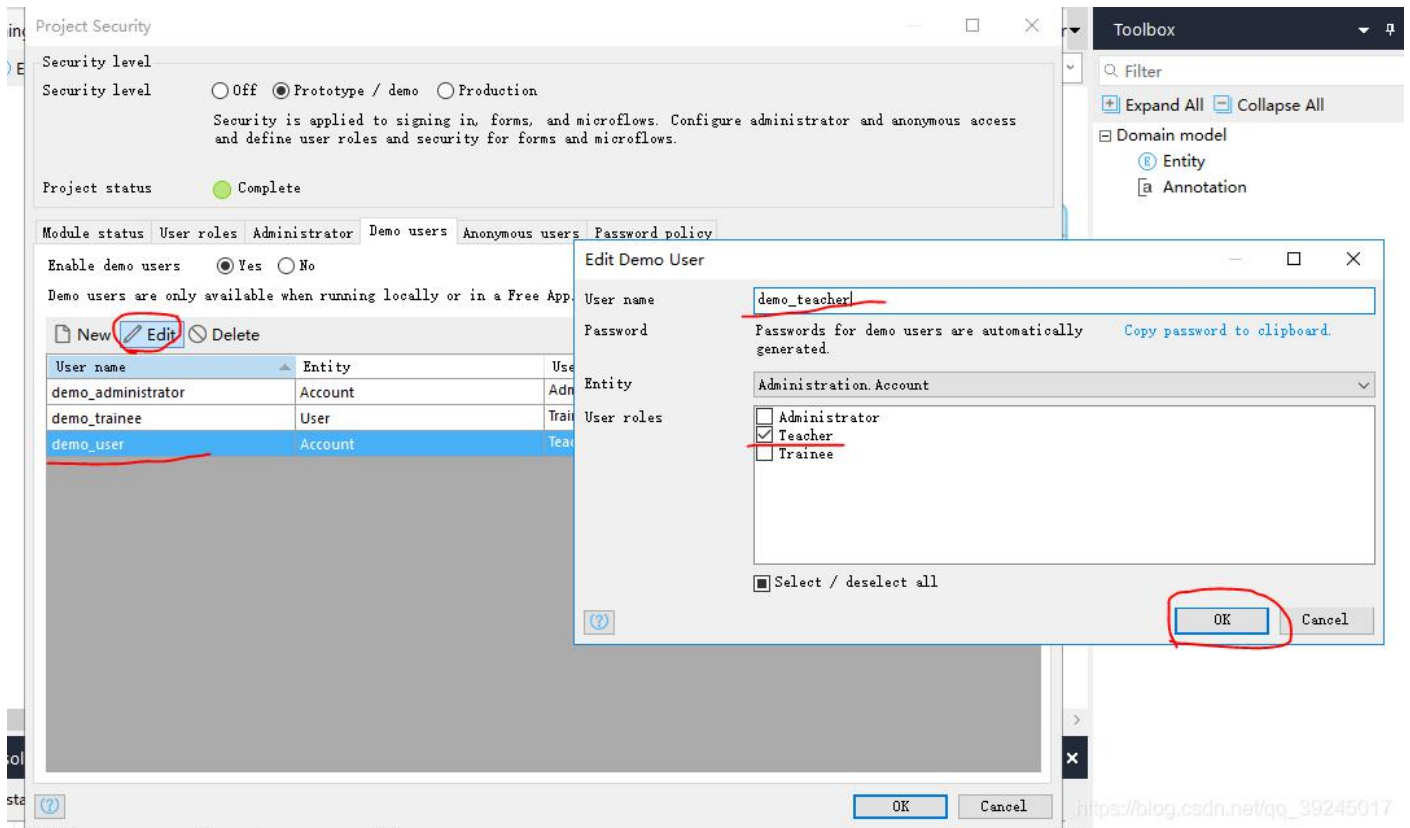
Here you can put a paragraph as subtitle

Add

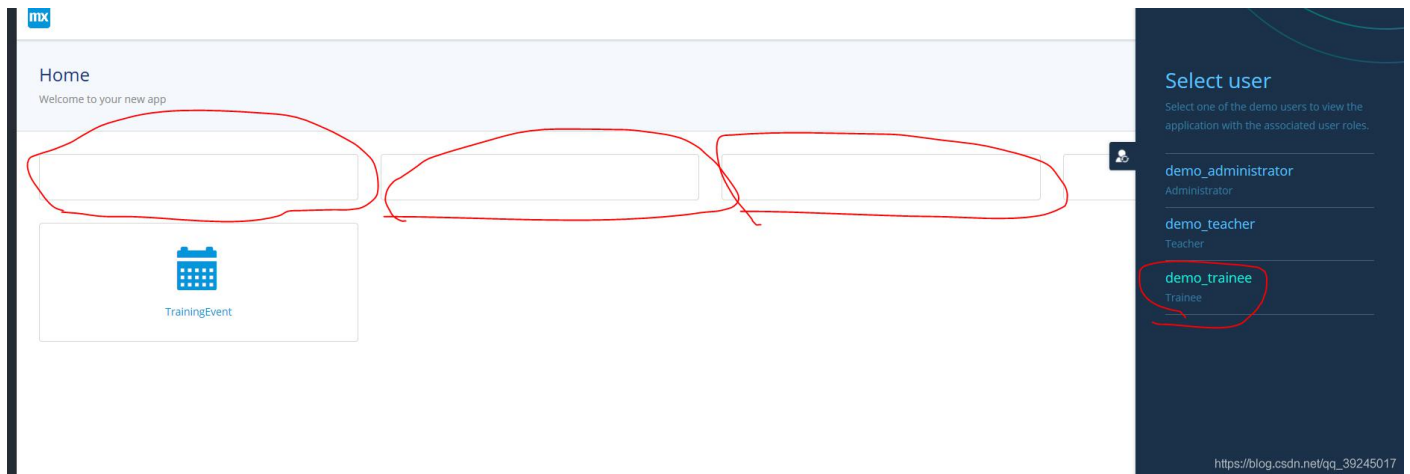
18. 经过以上配置，我们可以发现还需要设置一个 Trainee 报名人员的用户角色，我们来设置一下。我们双击 Project Security 找到 Demo users(演示用户)的页面，点击 New 按钮，在弹出的页面输入 demo_trainee 然后勾选 Trainee 角色，点击 Ok 按钮完成用户创建。



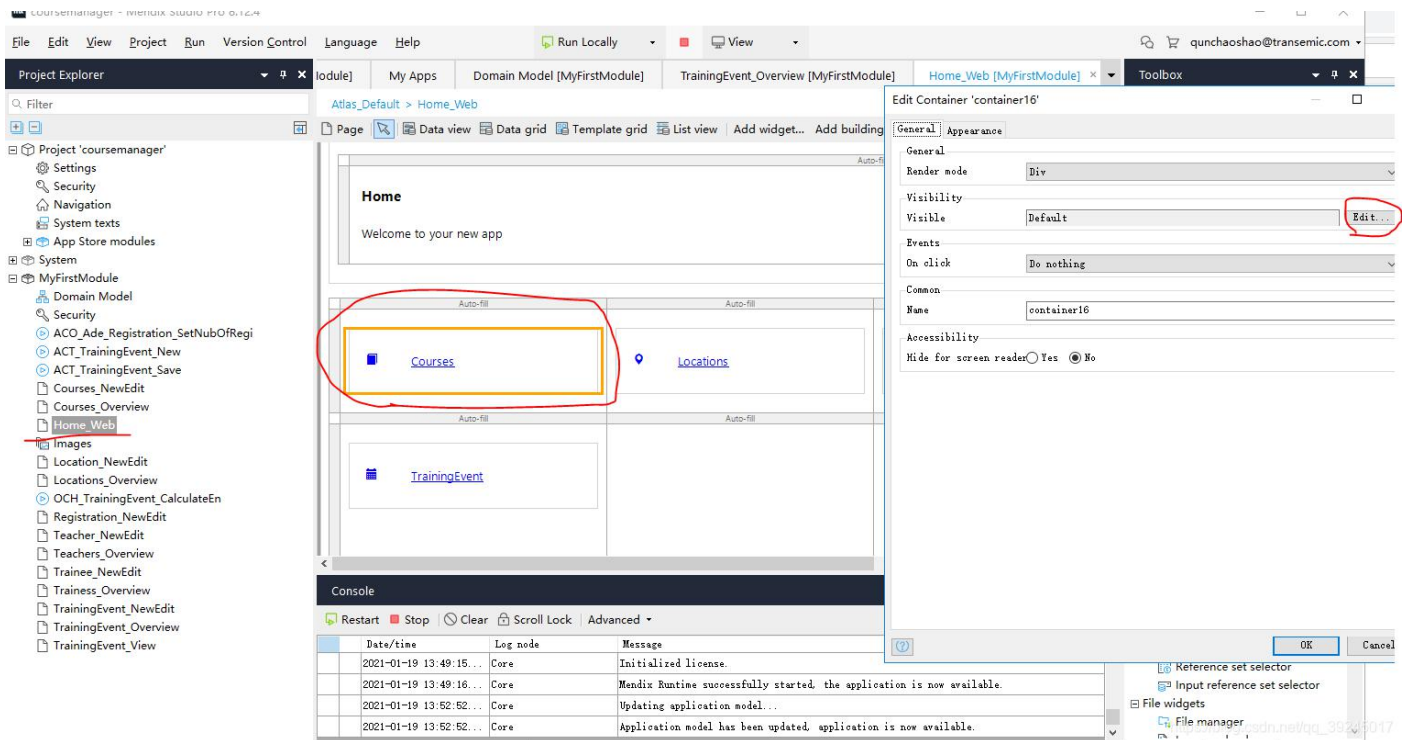
19. 我们将 demo_user 修改为 demo_teacher 方便理解。



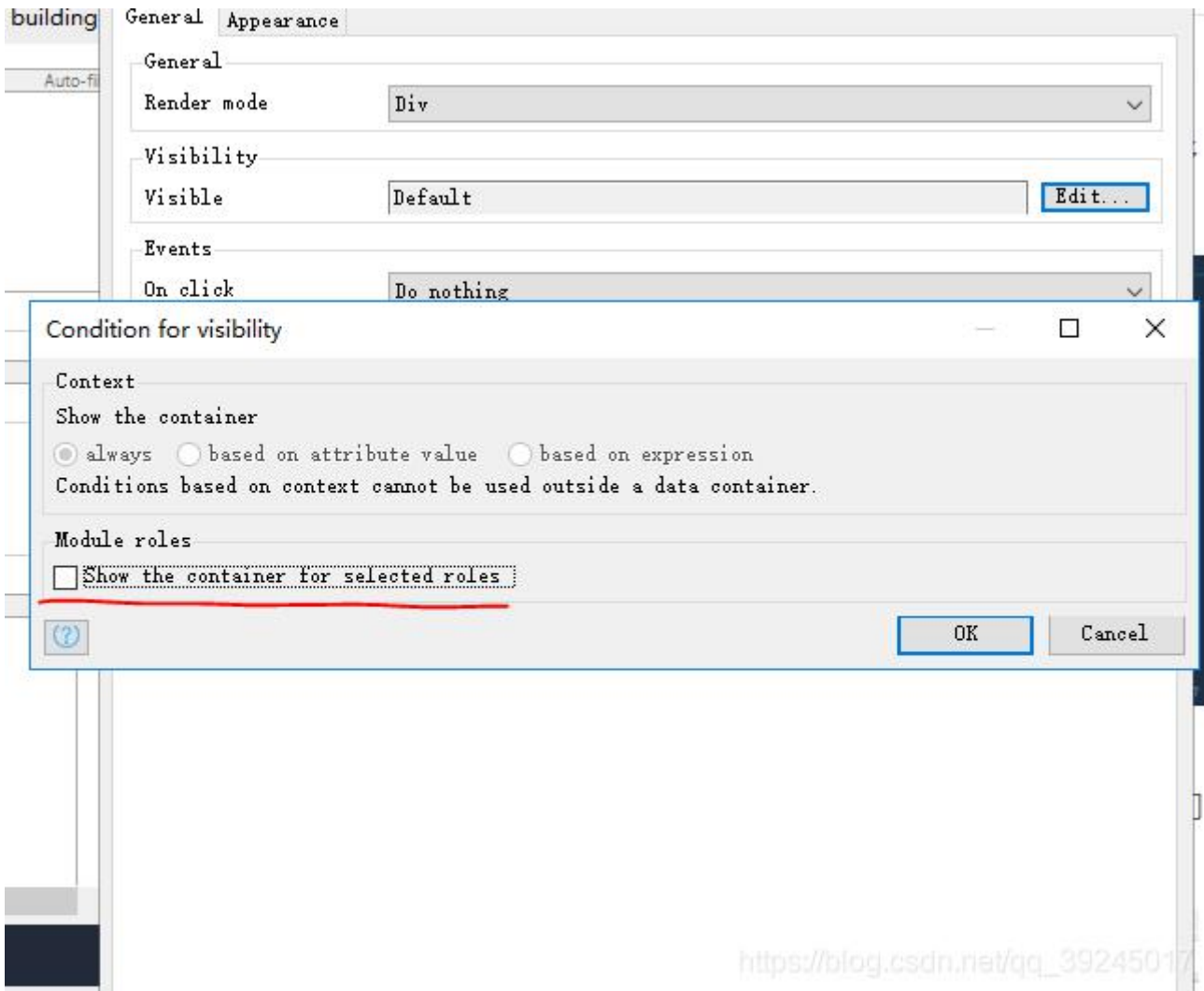
20. 保存所有操作，然后我们重新运行项目进行预览。选择 demo_trainee 用户，我们可以看到首页有很多的白框。很不美观。



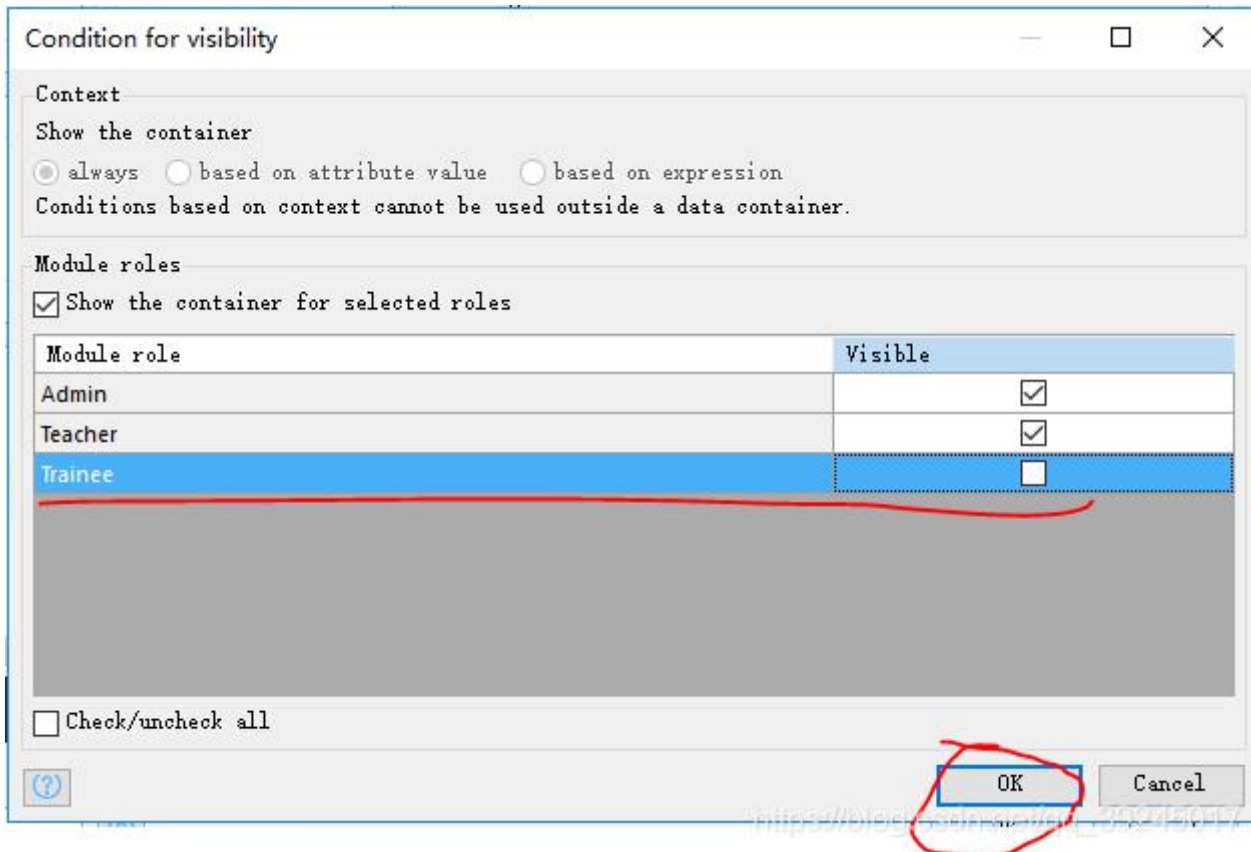
21. 那如何将白框跟角色进行绑定呢？我来操作一下。首先点击我们首页 Home_Web 页面。然后选中我们拖入的 Card 卡片下图红圈内，注意不要选错了。然后双击打开，在 visible 选项点击 Edit 进行编辑。



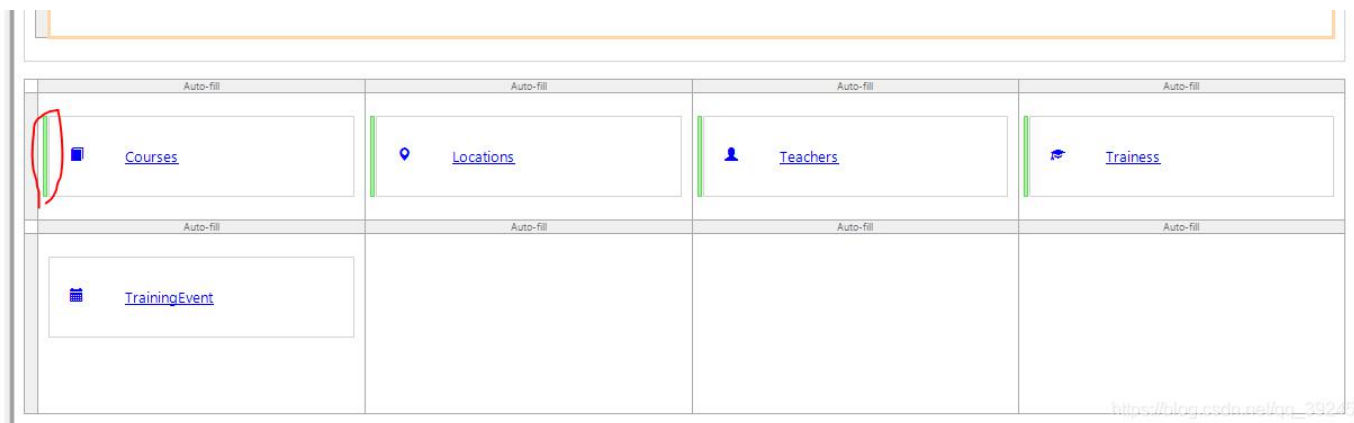
22. 在弹出的页面选择 Show the container for selected roles(只对勾选的角色进行展示)。



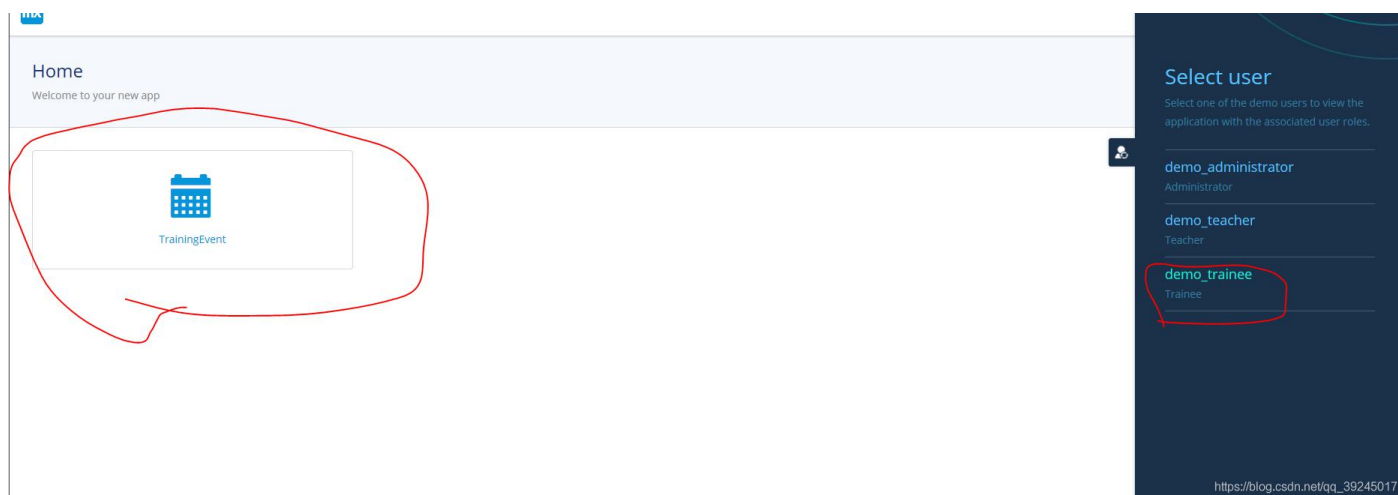
23. 然后将 Trainee 取消勾选显示，点击 Ok 就完成设置了。



24. 我们依次将其余的设置好，设置好以后会出现绿条。

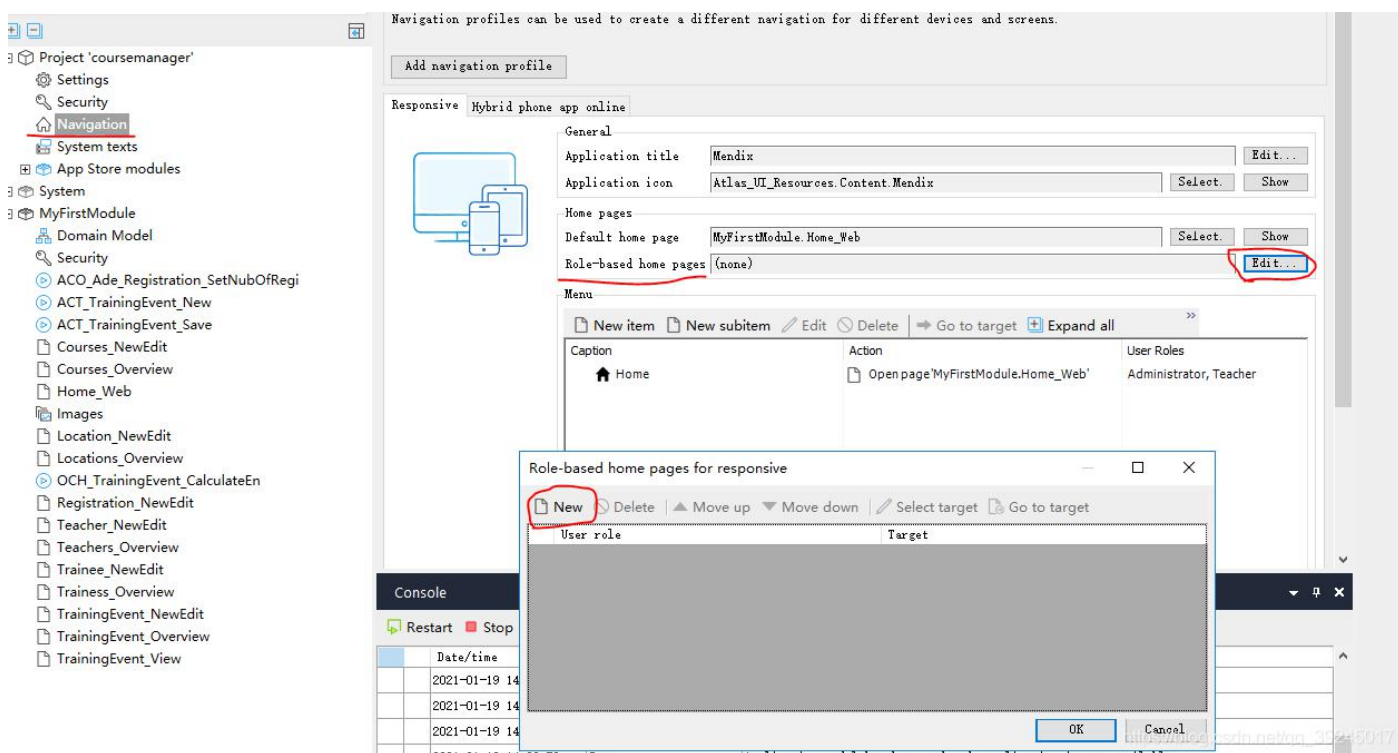


25. 然后我们重新运行项目进行预览。可以发现问题已经解决了。

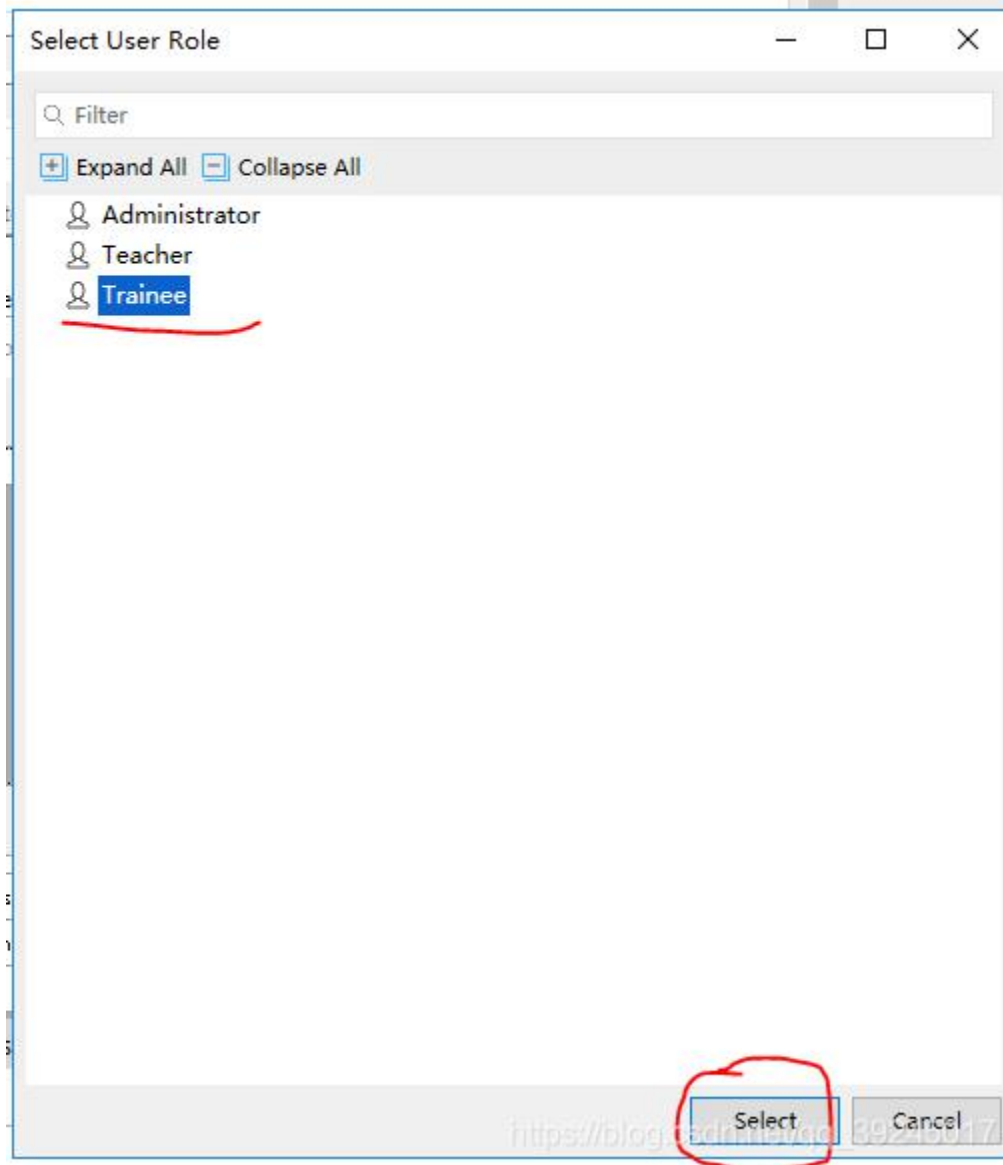


26. 我们发现其实报名人员只需要看到报名页面就行，不需要展示主页。那我来设置一下，点击

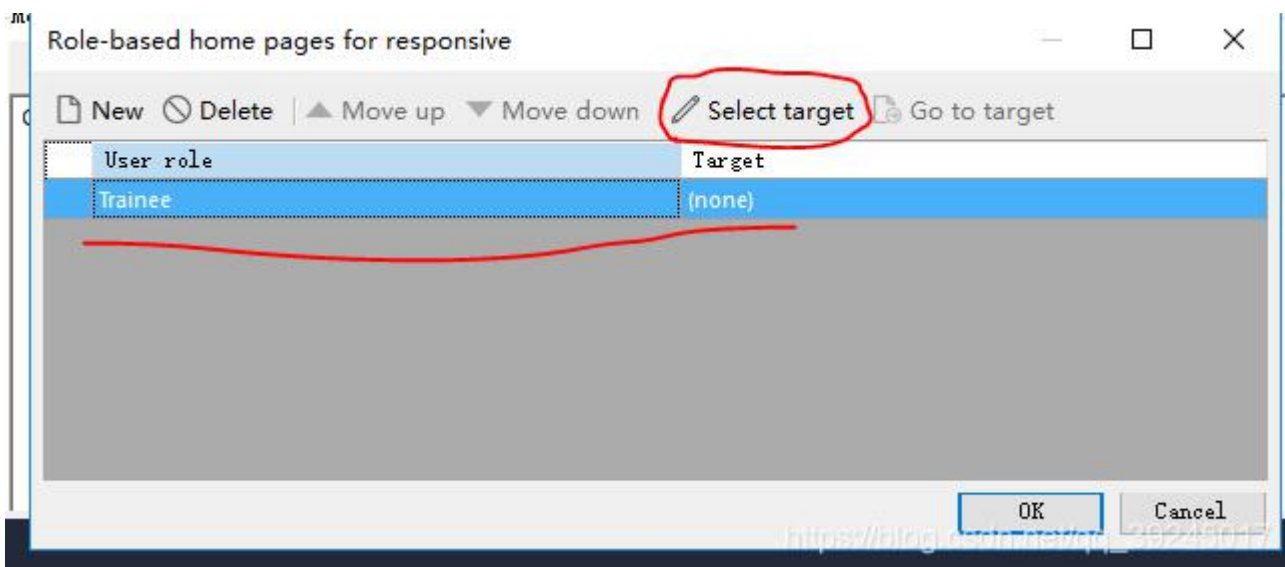
Navigation(系统设置),然后找到 Role-based home pages(根据不同角色展示页面)，然后点击 Edit 进行编辑，在弹出的页面选择 New 按钮新建规则。



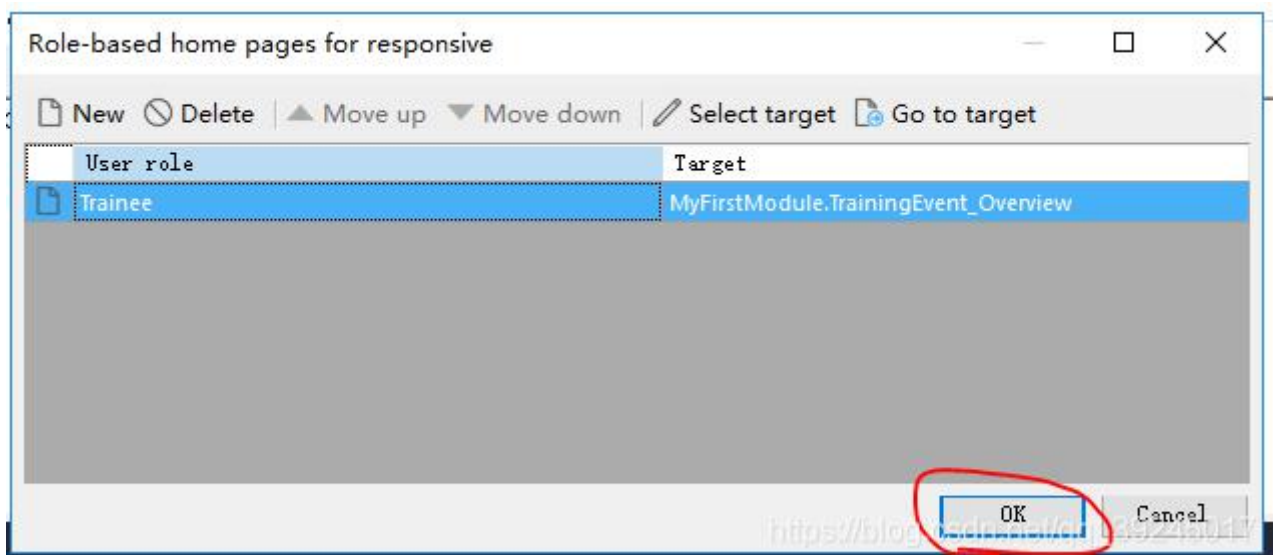
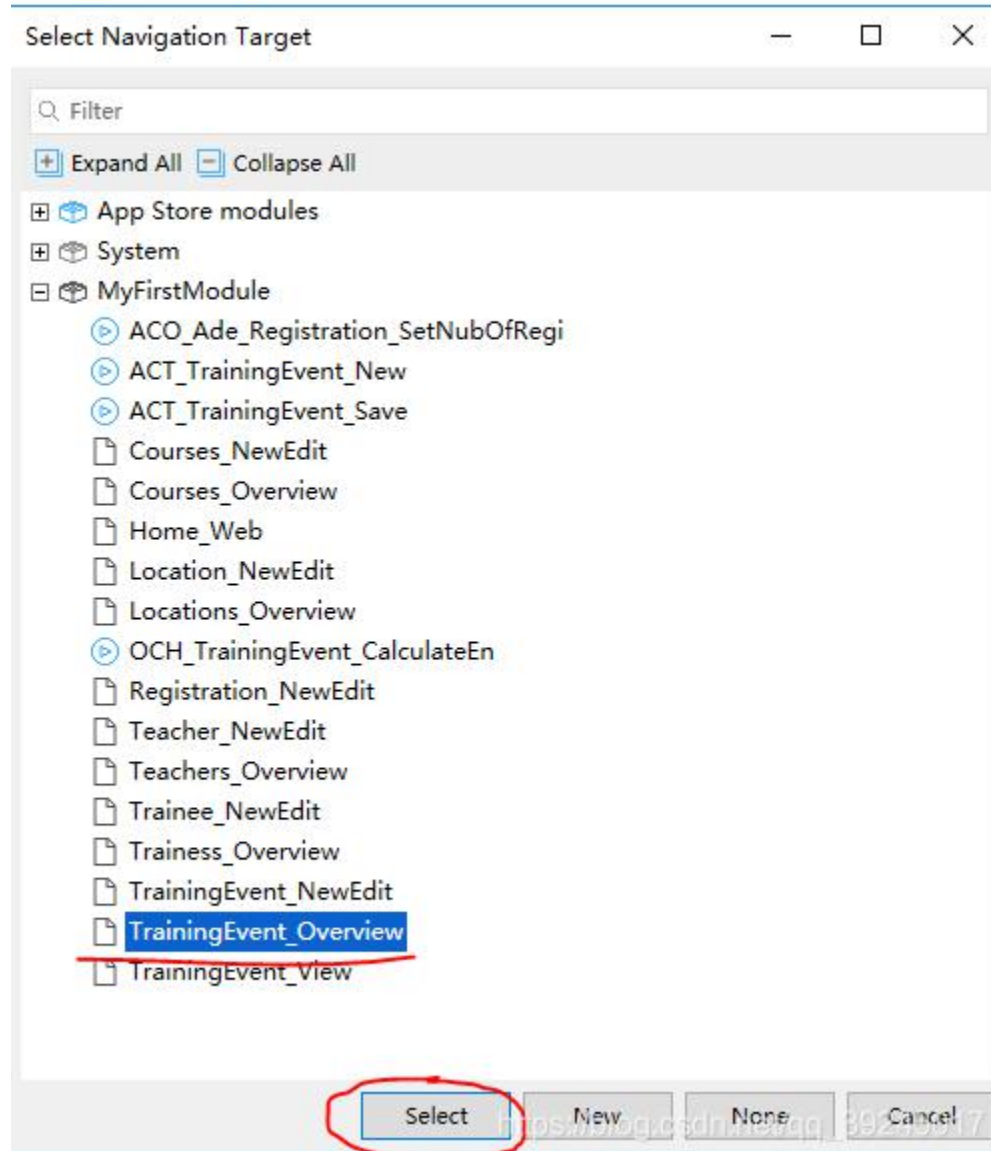
27. 然后在弹出的页面选择 Trainee 角色，然后选择 Select 按钮。



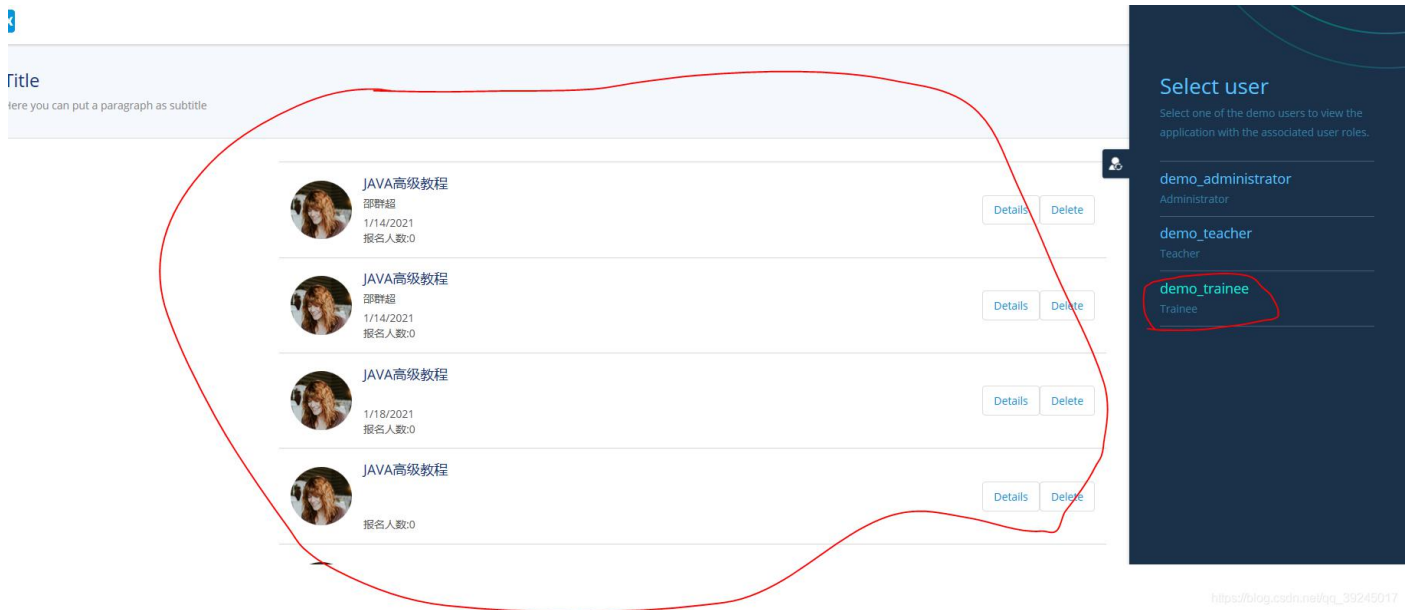
28. 然后选中新建的规则，双击或者点击 Select target 来选择展示的首页。



29. 然后我们选择活动报名页面，然后点击 Select->OK 按钮。完成创建



30. 然后我们重新运行项目进行预览。可以发现 demo_trainee 用户访问的直接是活动报名的页面。



5、 开启系统权限 Production

双击左侧目录树，点击 Project Security 模块。我们选择 Production(生产环境)开启系统的权限。当系统权限开启以后，每个模块下都会追加对数据模型的管理！其实该功能基于 **Prototype/demo** 上更

多的是为了开始对数据的保护。

Project Security

Security level

Security level

☐ Off
 ☐ Prototype / demo
 ☒ Production

Full security is applied. Configure administrator and anonymous access and define user roles and security for forms, microflows, entities, and reports.

Check security

☒ Yes
 ☐ No

If there are no other errors, Mendix Studio Pro checks per user role whether forms that are accessible for a certain role only refer to attributes and associations that are accessible for that same role. This assumes that each user role is independent and that users do not need two or more roles to access functionality in your application.

Project status

☒ Incomplete

Module status

User roles

Administrator

Demo users

Anonymous users

Password policy

Edit module security

Module	Page access	Nanoflow access	Microflow access	OData access	REST access	Entity access	Data set access
MyFirstModule	Complete	Complete	Complete	Complete	Complete	Incomplete	Complete

?

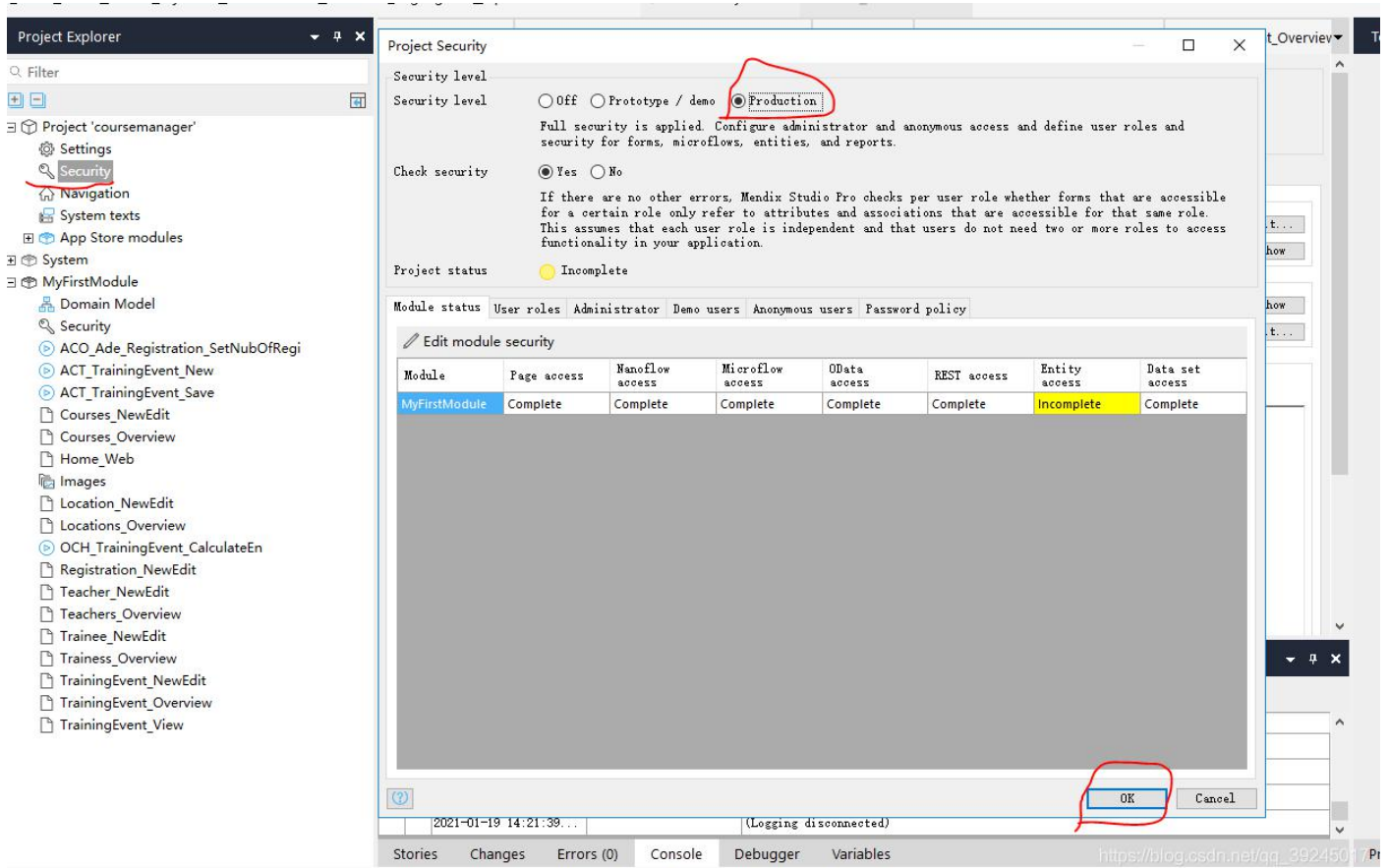
https://blog.cdn...

OK

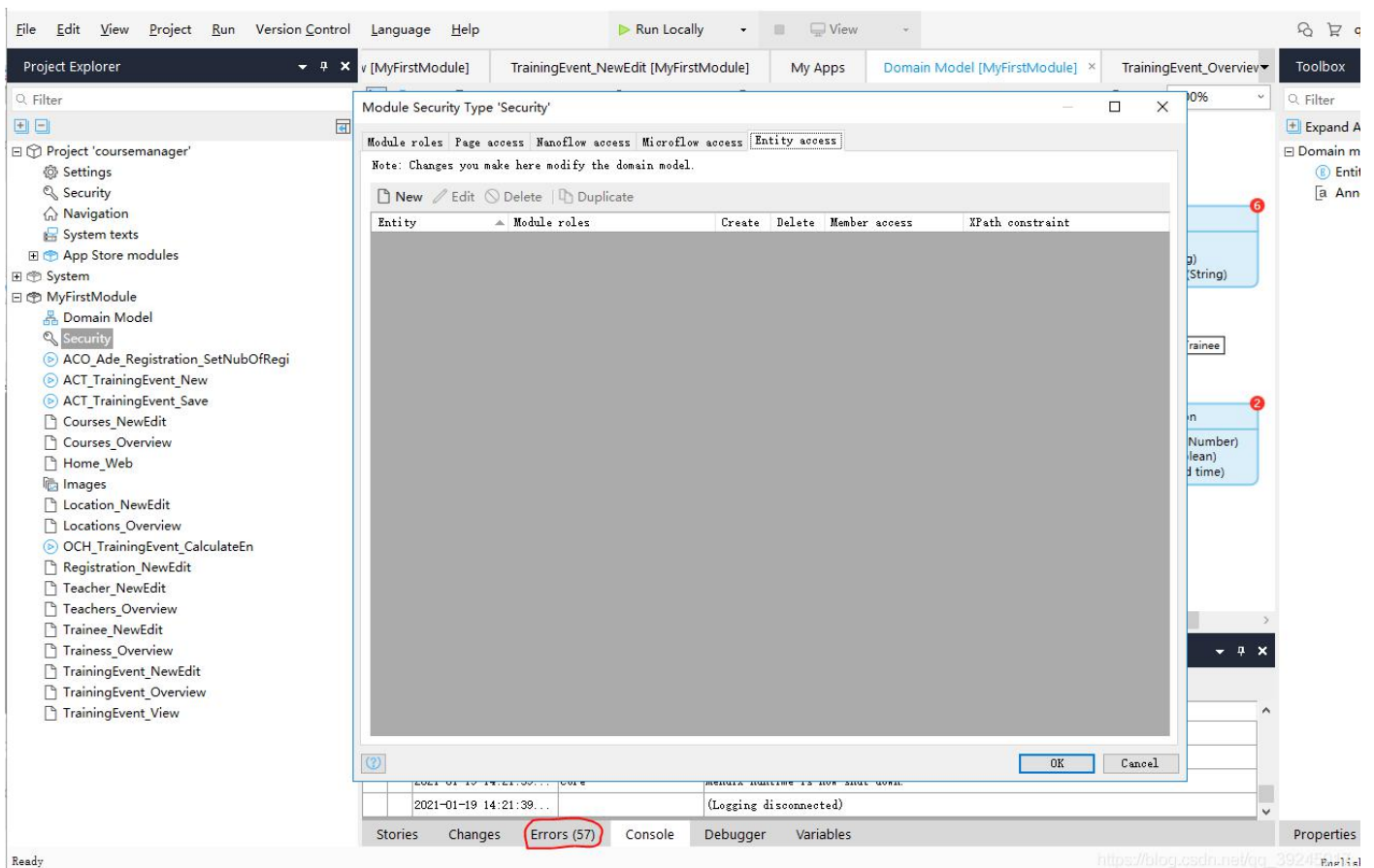
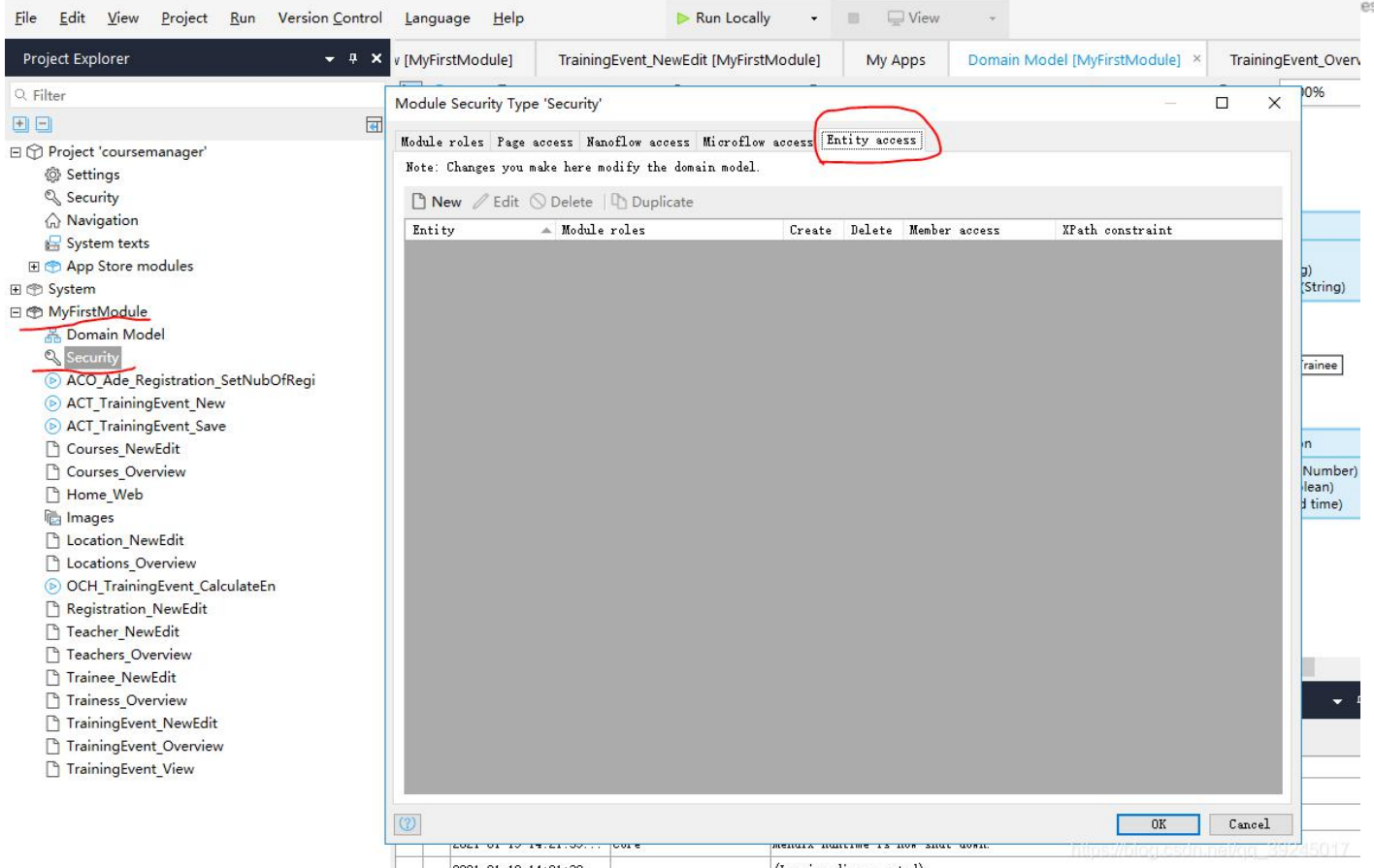
Cancel

5.1 应用场景

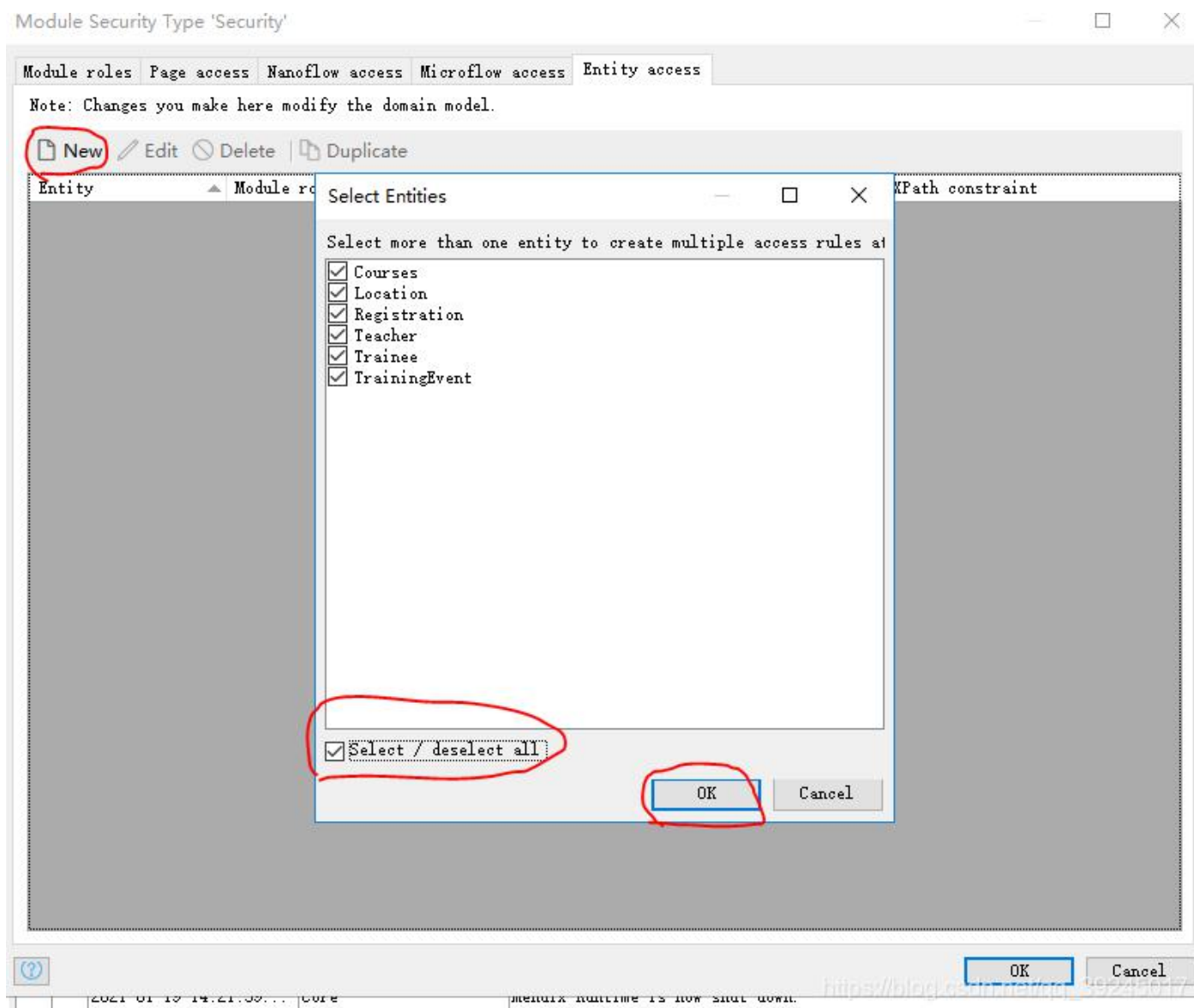
1. 点击左侧目录树中等个 Security，然后选择 Production 点击 Ok 按钮。打开 Production 权限。



2. 这时候你就会发现在你的模块权限中会增加 Entity access(数据模型)。同样的可以看到我们错误信息中也会有 57 个错误，是因为我们没有配置角色对数据模型的关系。所以会报错。

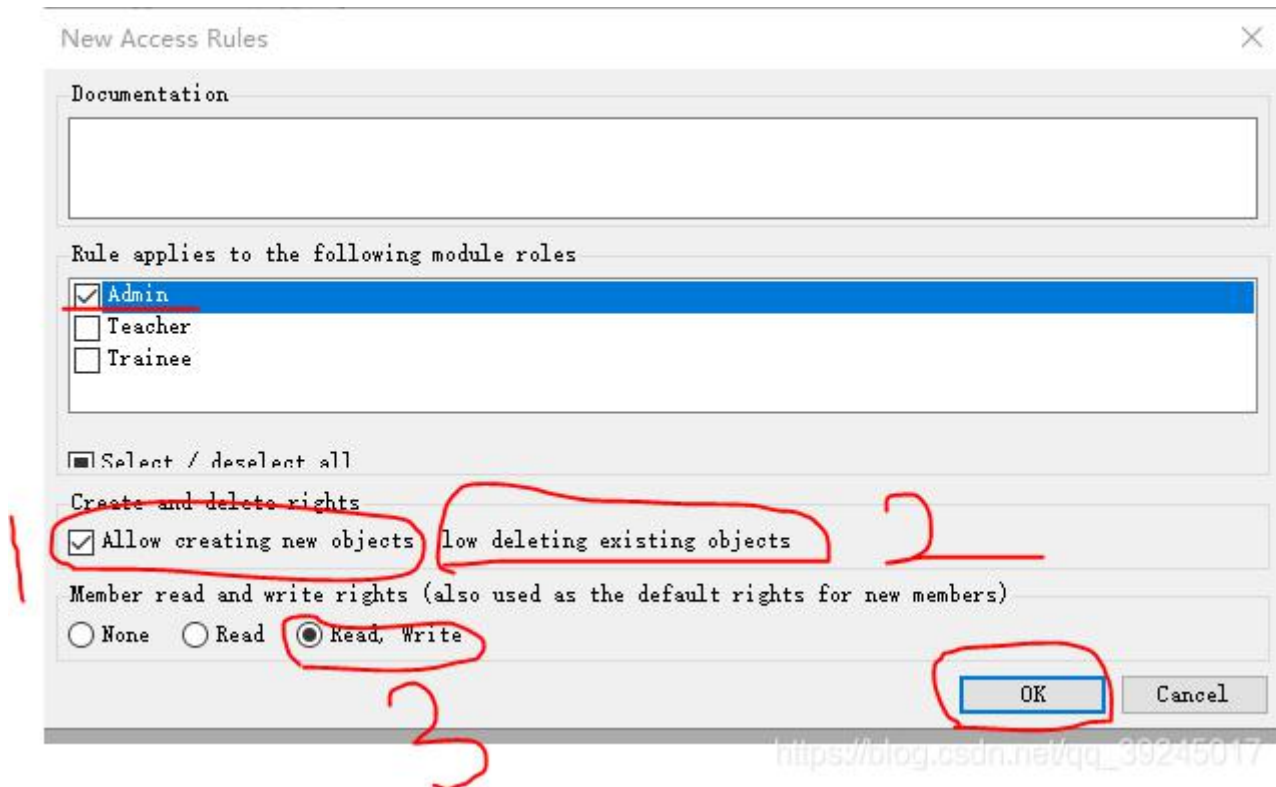


3. 我们现在来配置一下用户和数据模型之间的关系，我们先创建 Admin 的角色与数据模型的关系。点击 New 按钮，然后选择全部，然后点击 Ok 按钮。

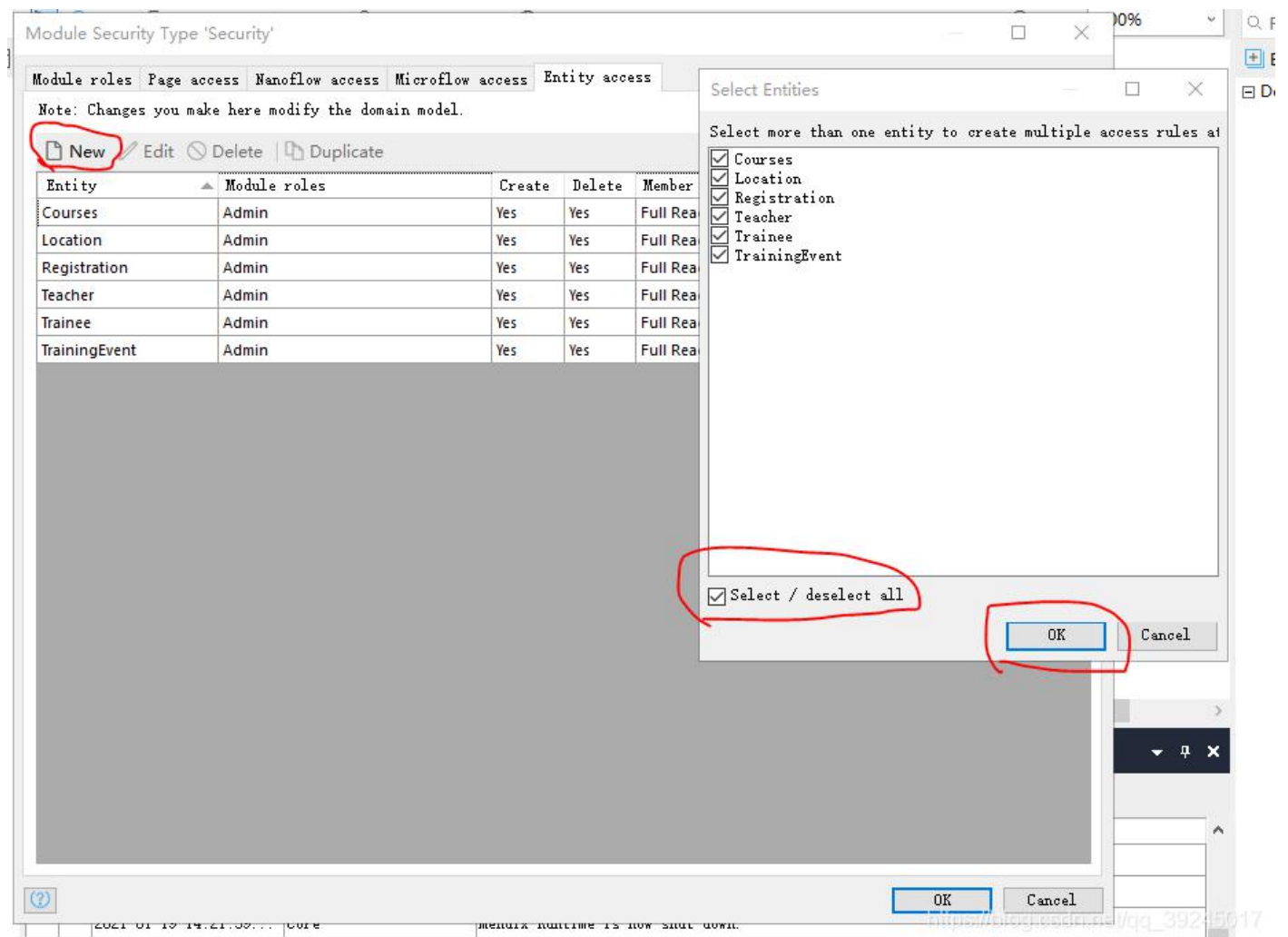


4. 在弹出的页面我们选择 Admin 角色，因为 Admin 角色是可以访问所有并有读写创建删除权限的，序号 1 是可以创建所有实体，序号 2 是可以删除所有实体(因为开发工具的 BUG 导致序号 2 展示不全，我们只需要点一下他就勾选了)，序号 3 代表的是读写权限(None 代表没有读写权限、Read 代表只读

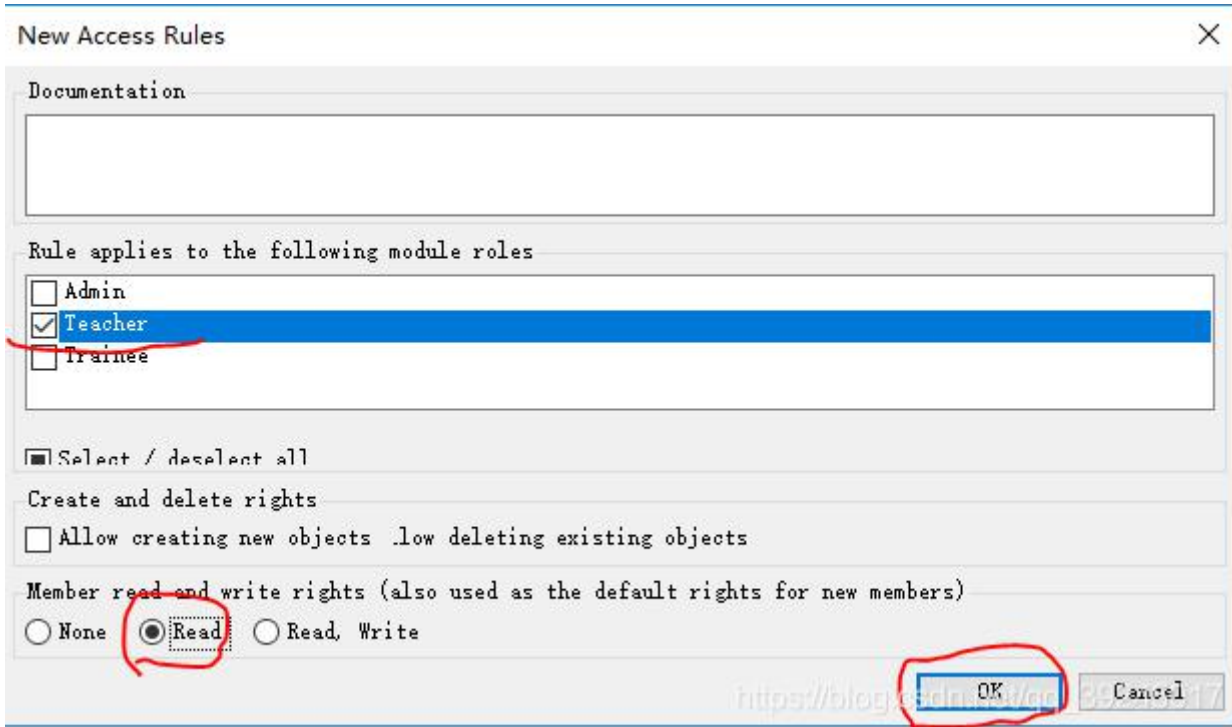
权限)然后单击 OK 按钮。完成此操作



5. 我们再点击一下 Teacher 老师的角色，点击 New 按钮，选择所有，然后点击 Ok 按钮。



6. 因为老师只有读的权限，没有写的权限，所以我们只勾选 Read(只读)然后点击 OK 按钮。



New Access Rules

Documentation

Rule applies to the following module roles

- ☐ Admin
- ☒ Teacher
- ☐ Trainee

☐ Select / deselect all

Create and delete rights

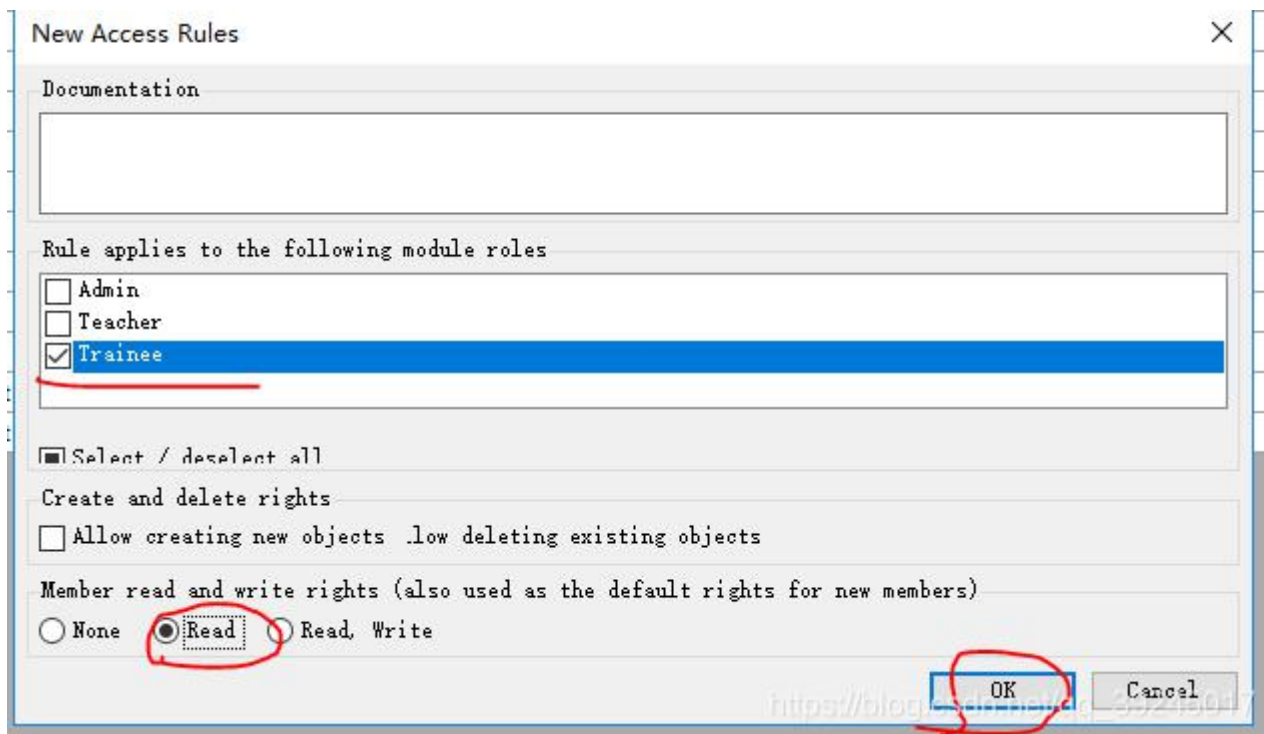
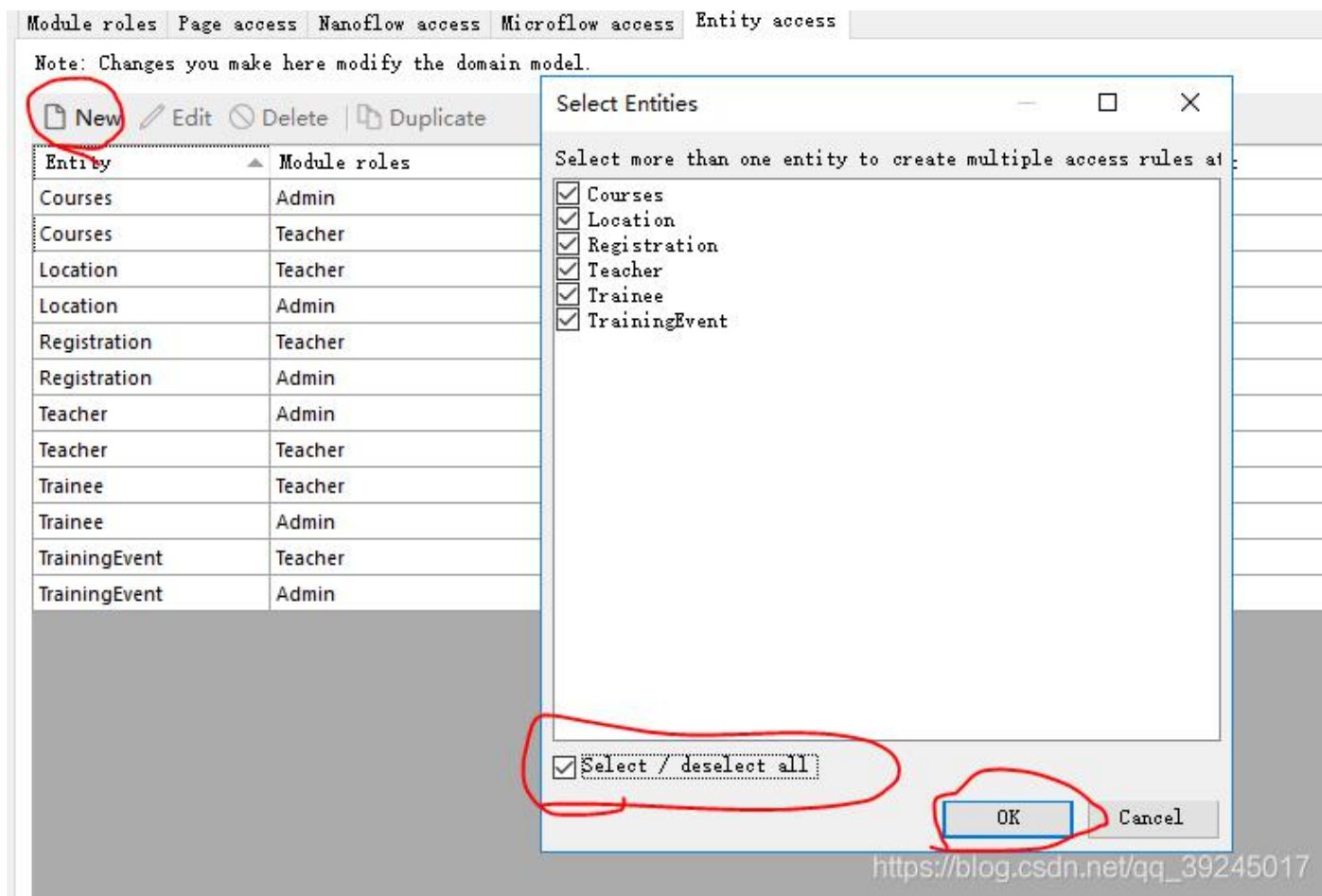
☐ Allow creating new objects .low deleting existing objects

Member read and write rights (also used as the default rights for new members)

☐ None ☒ Read ☐ Read, Write

https://blog.csdn.net/qq_3944317 OK Cancel

7. 学生也是只有读的权限，没有写的权限。所以跟老师的设置是一样的。

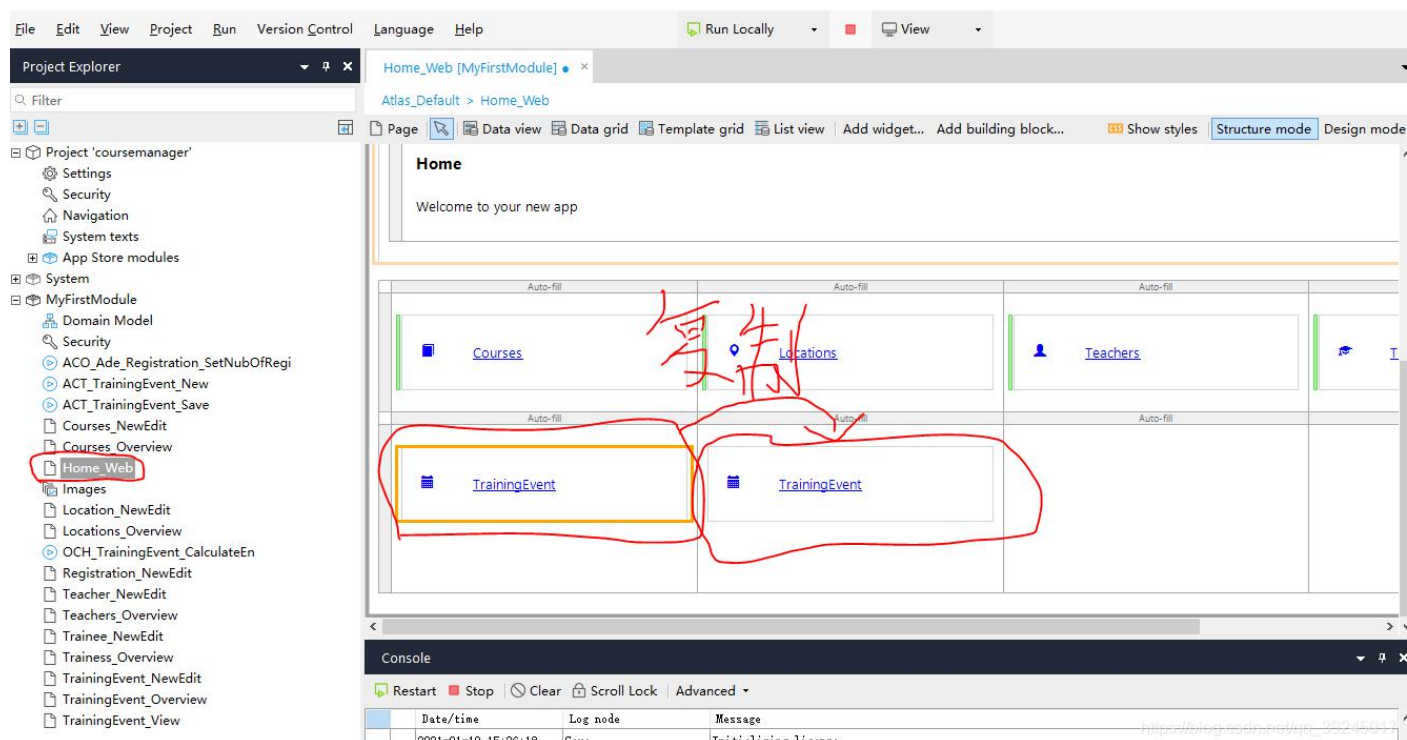


8. 然后我们重新运行项目进行预览。可以发现 demo_trainee 和 demo_teacher 用户访问的所有页面都没有新增和删除的操作了。只有 demo_administrator 有新增和删除的权限。

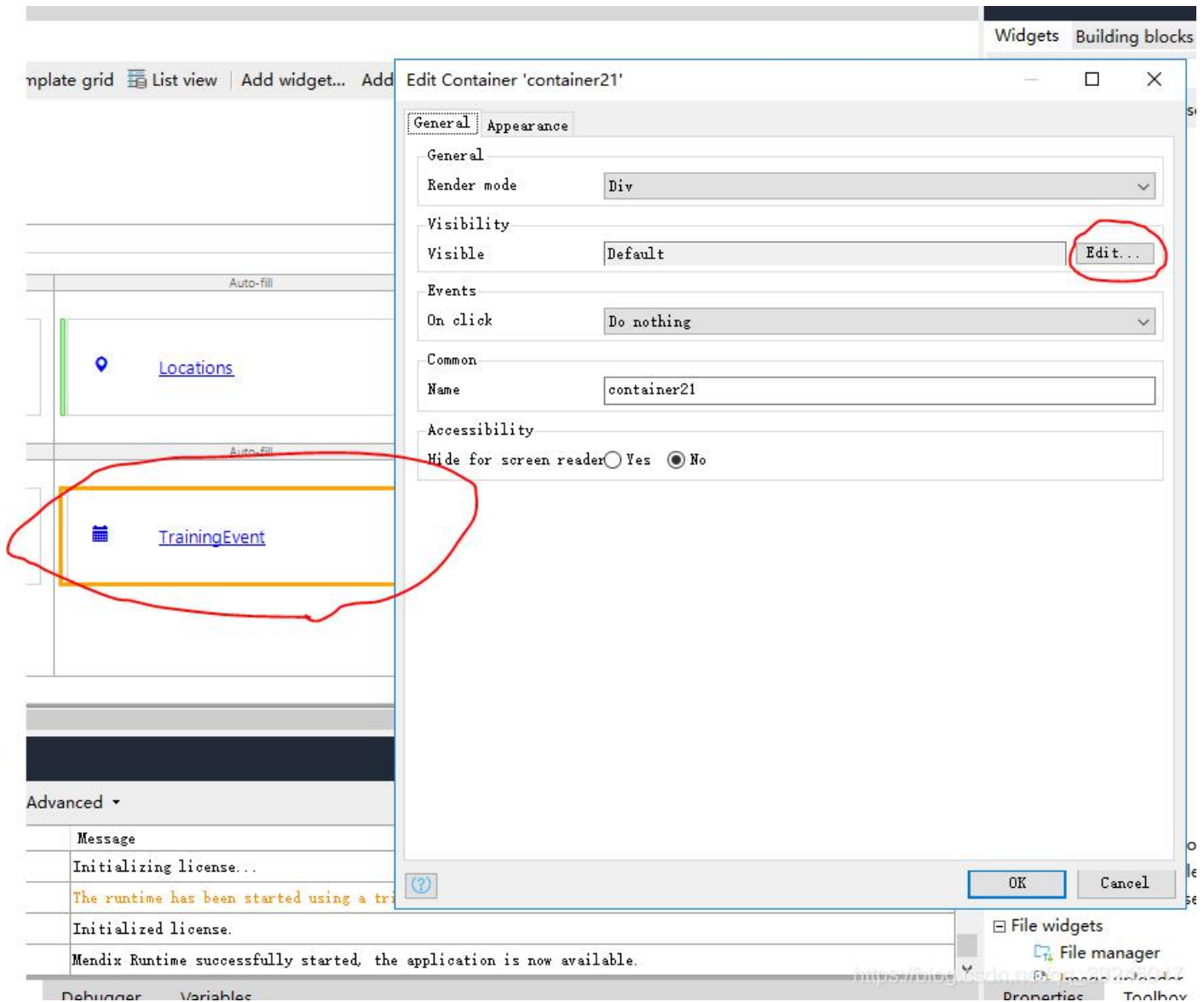
三、用户新增

经过以上操作我们发现，用户新增需要在开发工具中很麻烦，我们能不能在页面上新增用户呢？答案：可以的。Mendix 有一套自己的用户新增页面。我们可以直接拿过来用，很方便的。我开始操作一下。

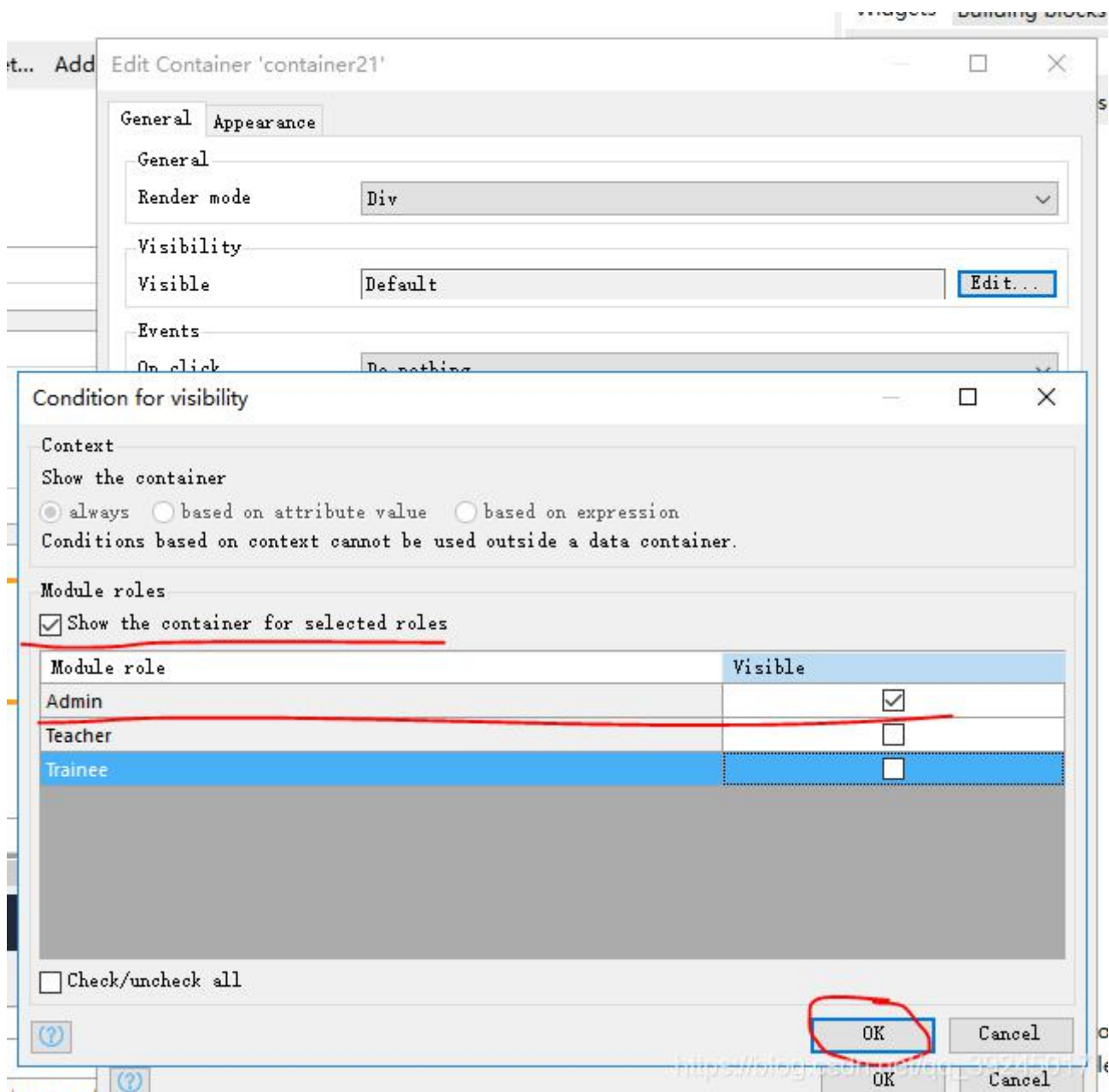
1. 首先找到我们程序的 Home_Web(主页面)，然后选中 TrainingEvent 的 card(卡片)组件，复制，粘贴。并双击打开。



2. 我们双击刚复制的 Card(卡片), 然后点击编辑 Edit, 让他只能管理员看见。



3. 选择 Show the container for selected roles, 然后只让管理员看见, 然后点击 OK->Ok 按钮。



Edit Container 'container21'

General Appearance

General

Render mode Div

Visibility

Visible based on selected roles Edit...

Events

On click Do nothing

Common

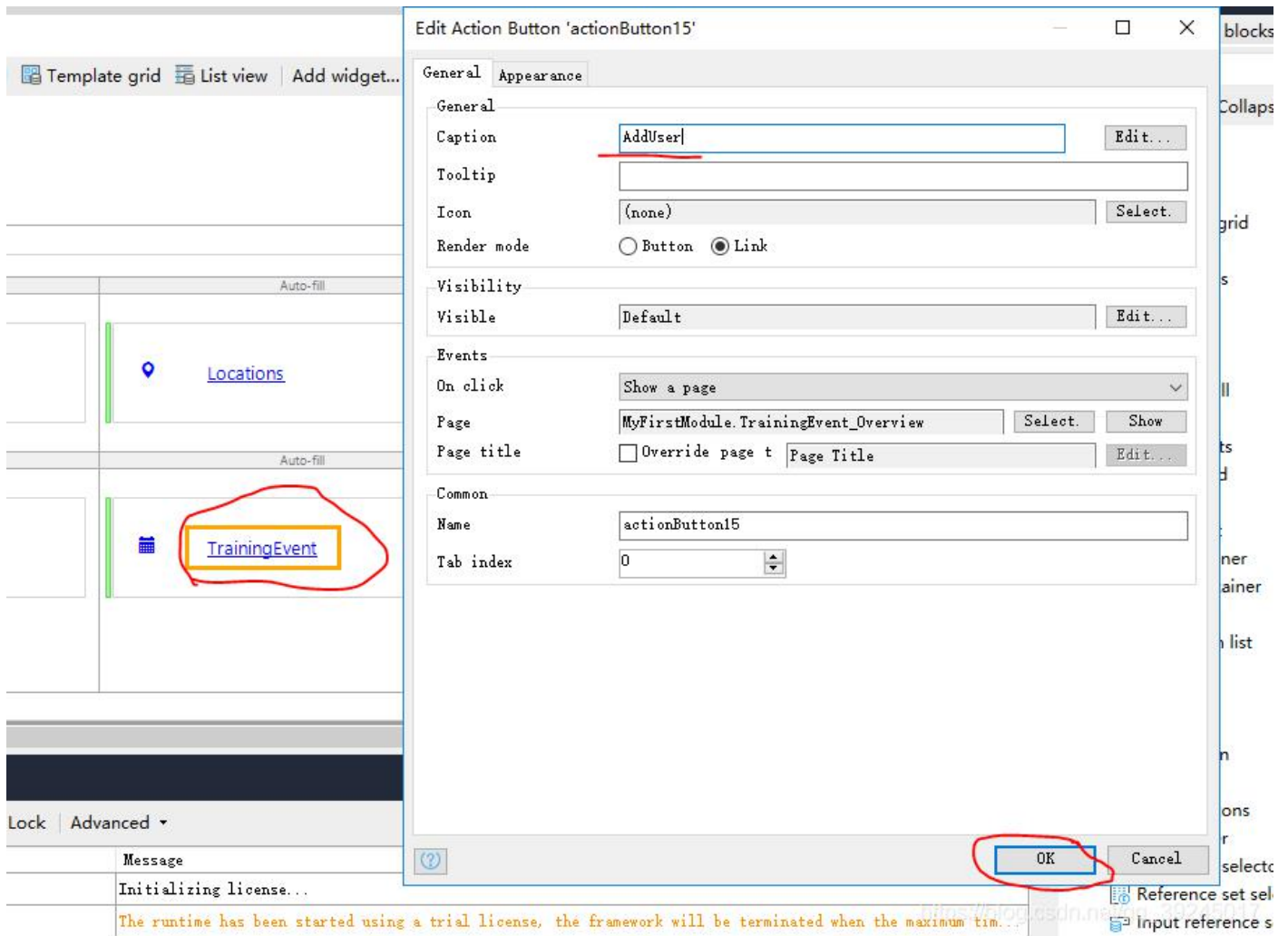
Name container21

Accessibility

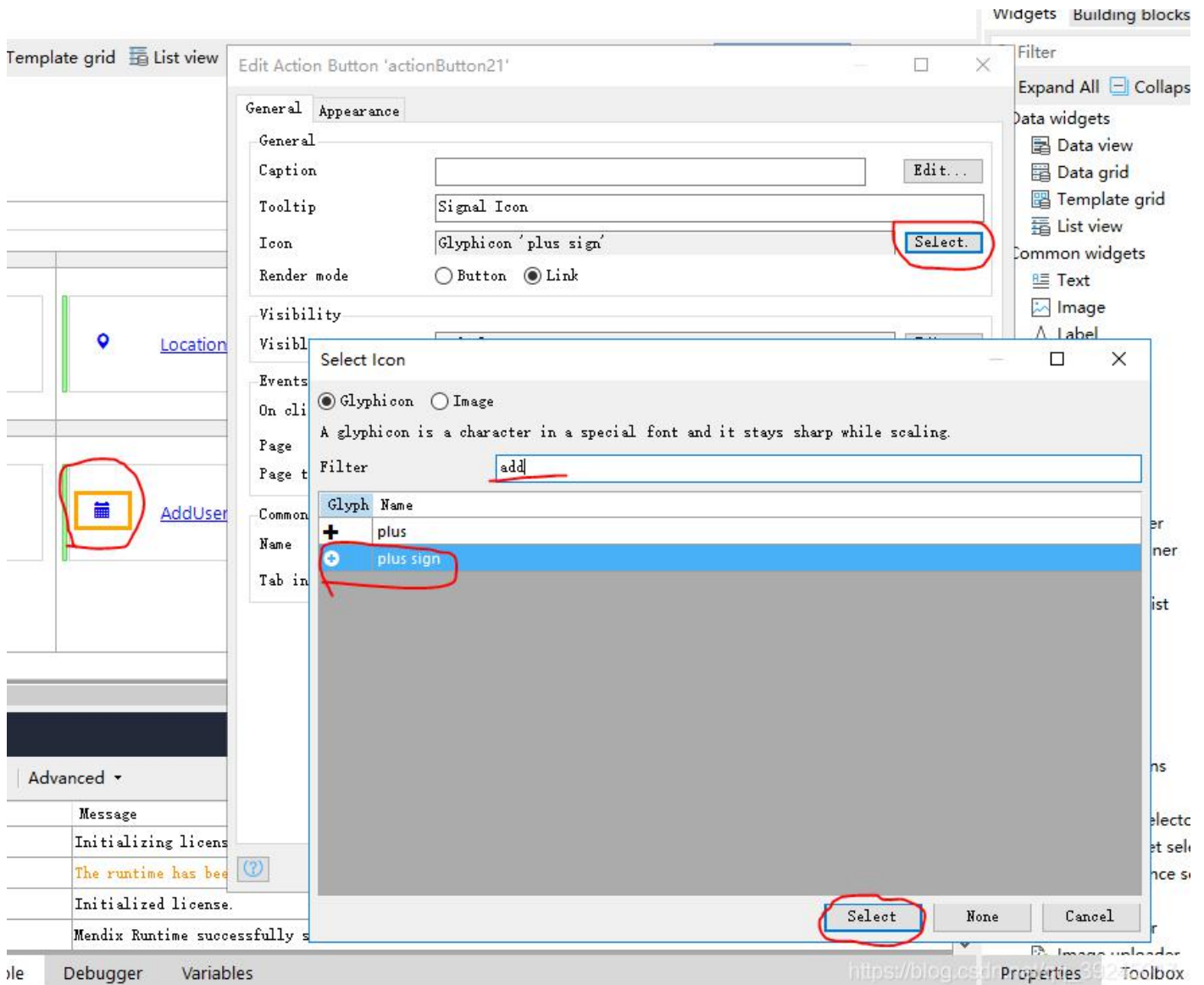
Hide for screen reader ☐ Yes ☒ No

<https://blog.mendix.com/...> OK Cancel

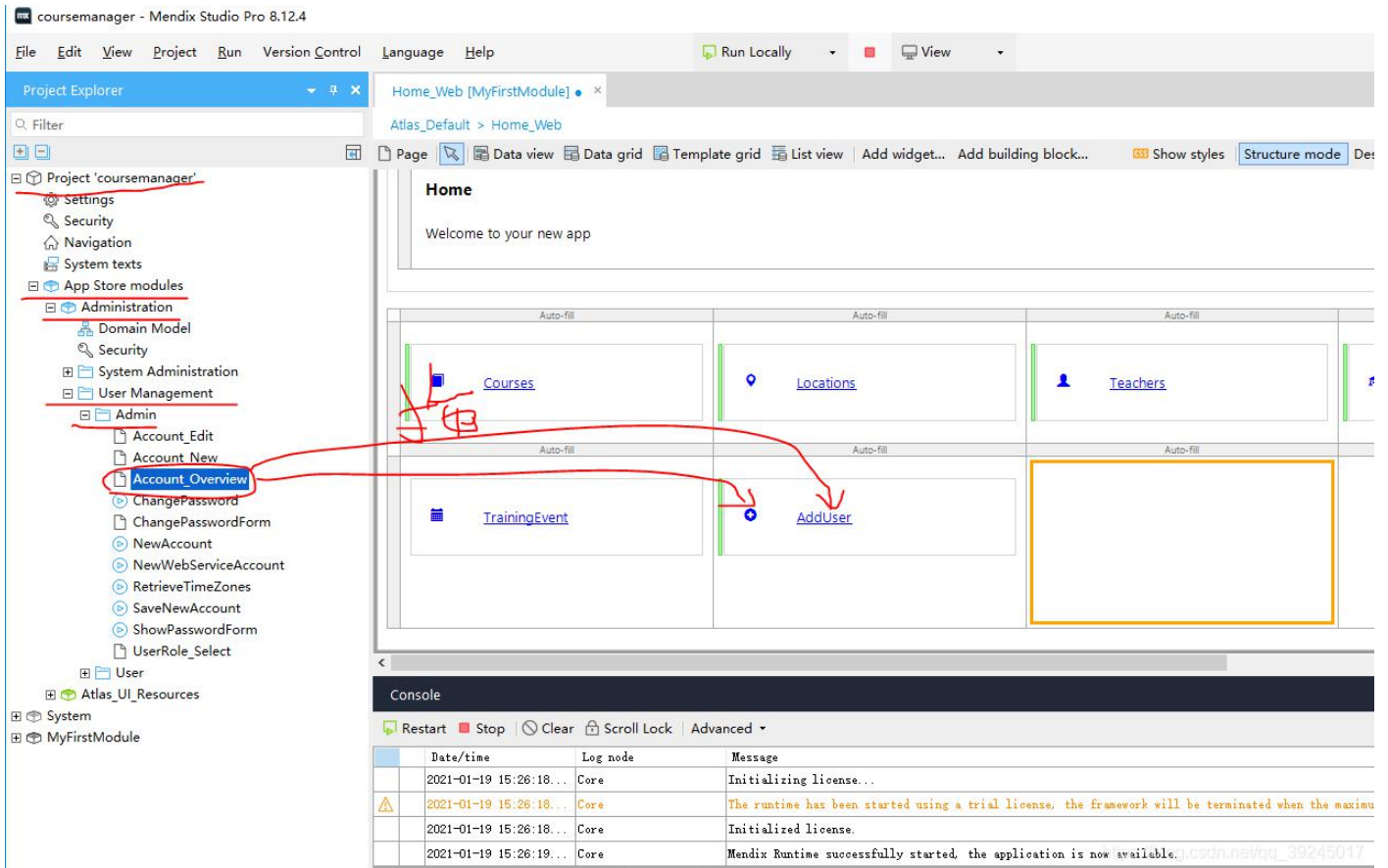
4. 然后我们修改名称为 AddUser。



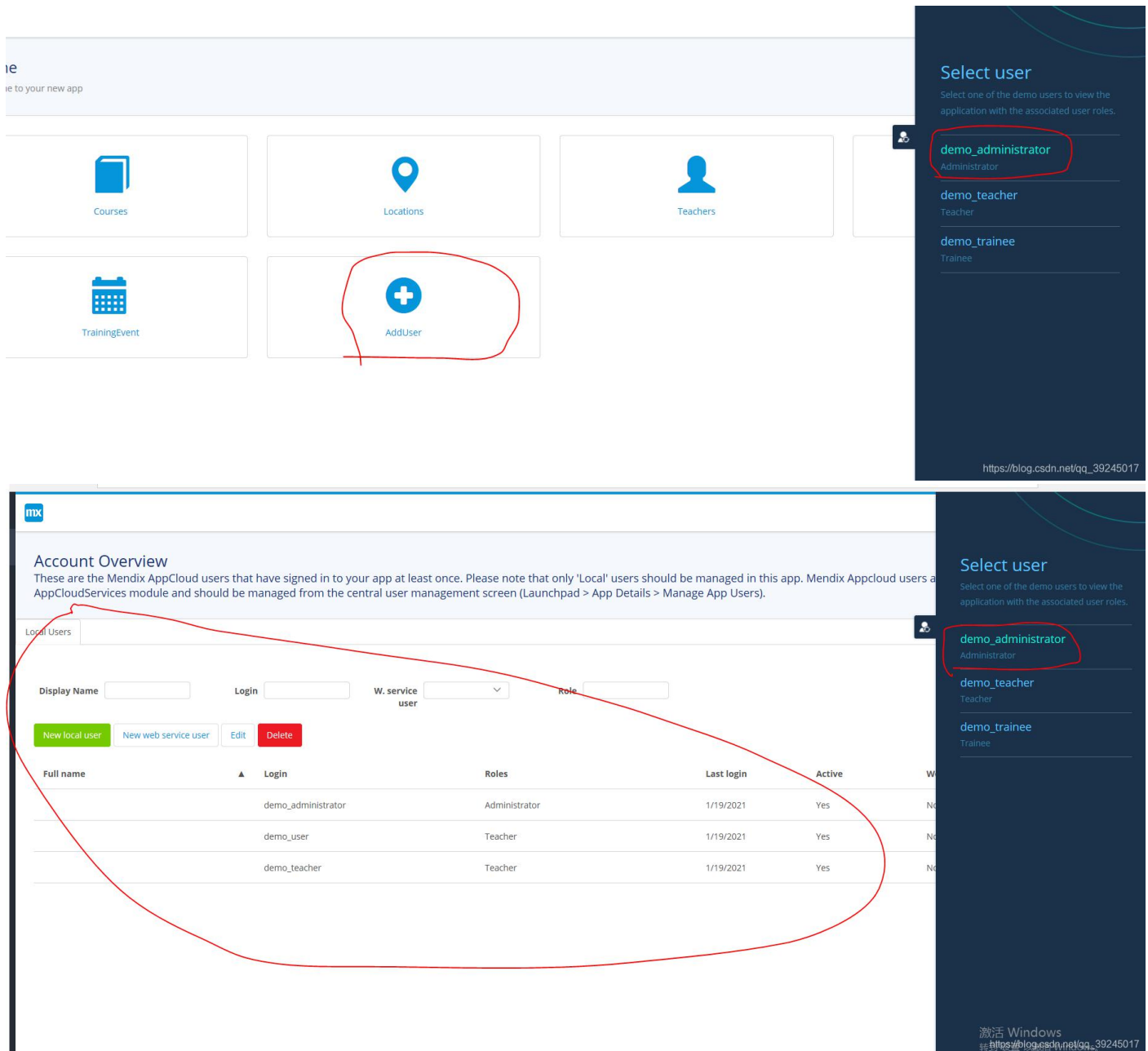
5. 修改图标。



6. 以上操作修改完毕以后。开始我们的最关键一步，也是最简单的一步。看好了各位！！找到对应目录下的 Account_Overview(用户管理页面)，直接拖到图标和文字中。



7. 然后我们重新运行项目进行预览。可以发现可以新增用户了。该功能大功告成。



四、程序退出功能

1. 打开指定的目录，然后点击下图红圈内容(New item)创建一个导航页，然后输入名称为 Sign Out，点击 Select 我们选择一个图标。

File Edit View Project Run Version Control Language Help Run Locally View

Project Explorer

- Filter
- Project 'coursemanager'
- Settings
- Security
- Navigation
- System texts
- App Store modules
- Administration
 - Domain Model
 - Security
- System Administration
- User Management
- Admin
 - Account_Edit
 - Account_New
 - Account_Overview
 - ChangePassword
 - ChangePasswordForm
 - NewAccount
 - NewWebServiceAccount
 - RetrieveTimeZones
 - SaveNewAccount
 - ShowPasswordForm
 - UserRole_Select
- User
- Atlas_UI_Resources
- System
- MyFirstModule

Home_Web [MyFirstModule] Navigation

Profiles

Navigation profiles can be used to create a different navigation for different devices and screens.

Add navigation profile

Responsive Hybrid phone app online

General

Application title: Mendix Edit...

Application icon: Atlas_UI_Resources.Content.Mendix Select Show

Home pages

Default home page: MyFirstModule.Home_Web Select Show

Role-based home pages: Trainee Edit...

Menu

New item New subitem Edit Delete Go to target Expand all Collapse all

Caption: Home Action: Open page 'MyFirstModule.Home_Web' User Roles: Administrator, Teacher

New Menu Item

General

Caption: Sign Out Select

Icon: (none) Select

On click: Do nothing

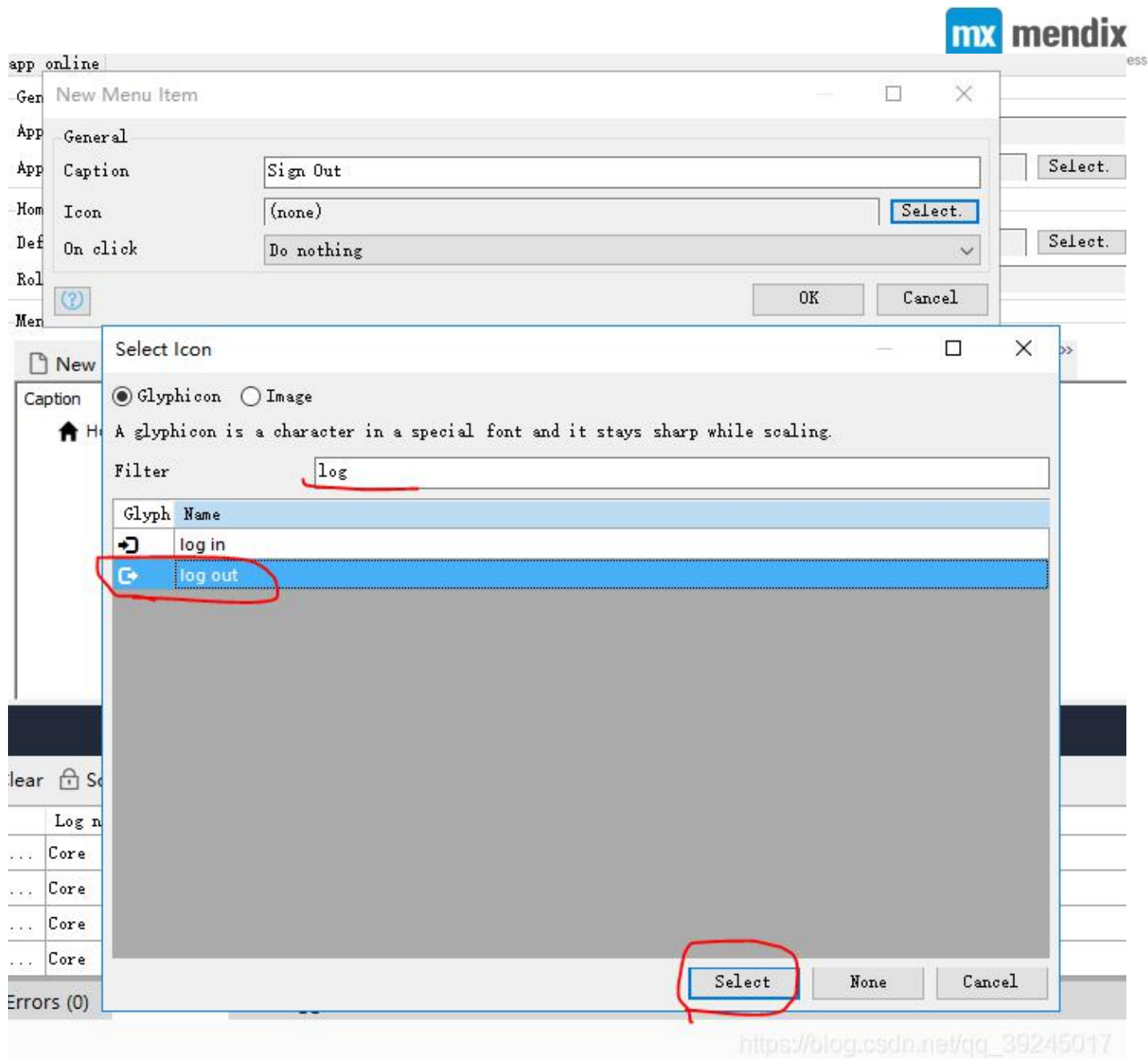
OK Cancel

Console

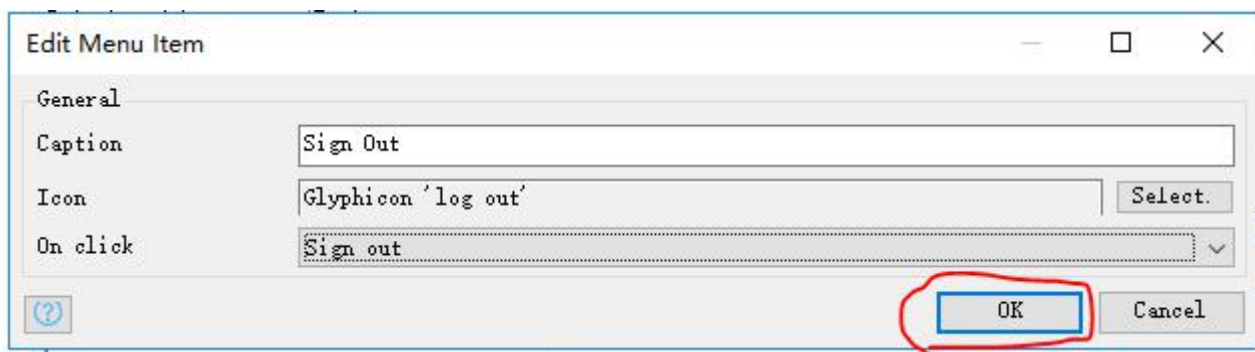
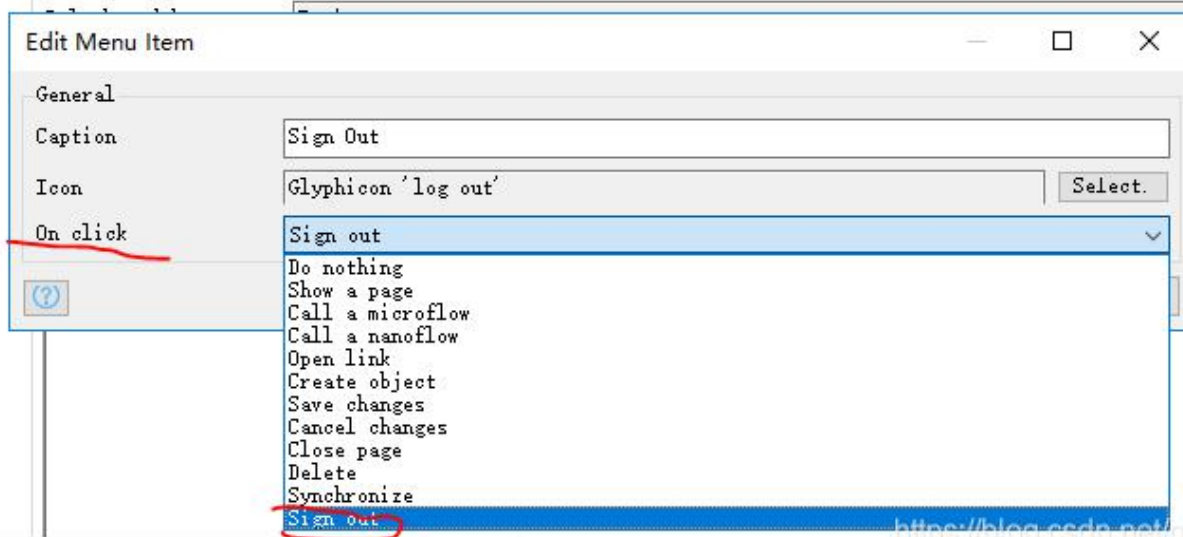
Restart Stop Clear Scroll Lock Advanced

Date/time	Log node	Message
2021-01-19 15:26:18...	Core	Initialized license.
2021-01-19 15:26:19...	Core	Mendix Runtime successfully started, the application is now available.
2021-01-19 16:38:45...	Core	Updating application model...

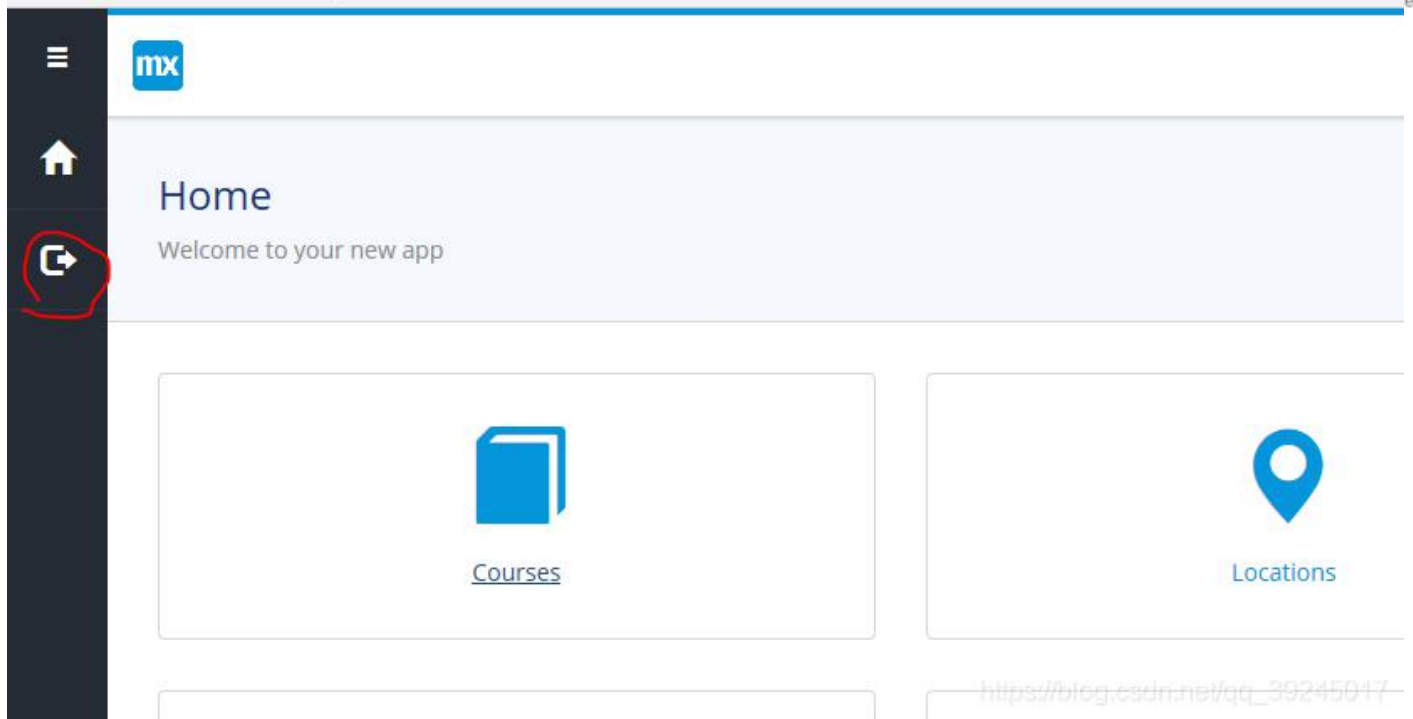
https://blog.csdn.net/qn_51245917



2. 选择好图标以后我们选择一个事件，系统给我们提供了一个 Sign Out 退出事件，所以我们直接选他就可以了，点击 Ok 按钮保存操作。



3. 然后我们重新运行项目进行预览。可以发现退出导航栏了，点击按钮系统会自动的跳转到我们的登陆页面。



Sign in

User name

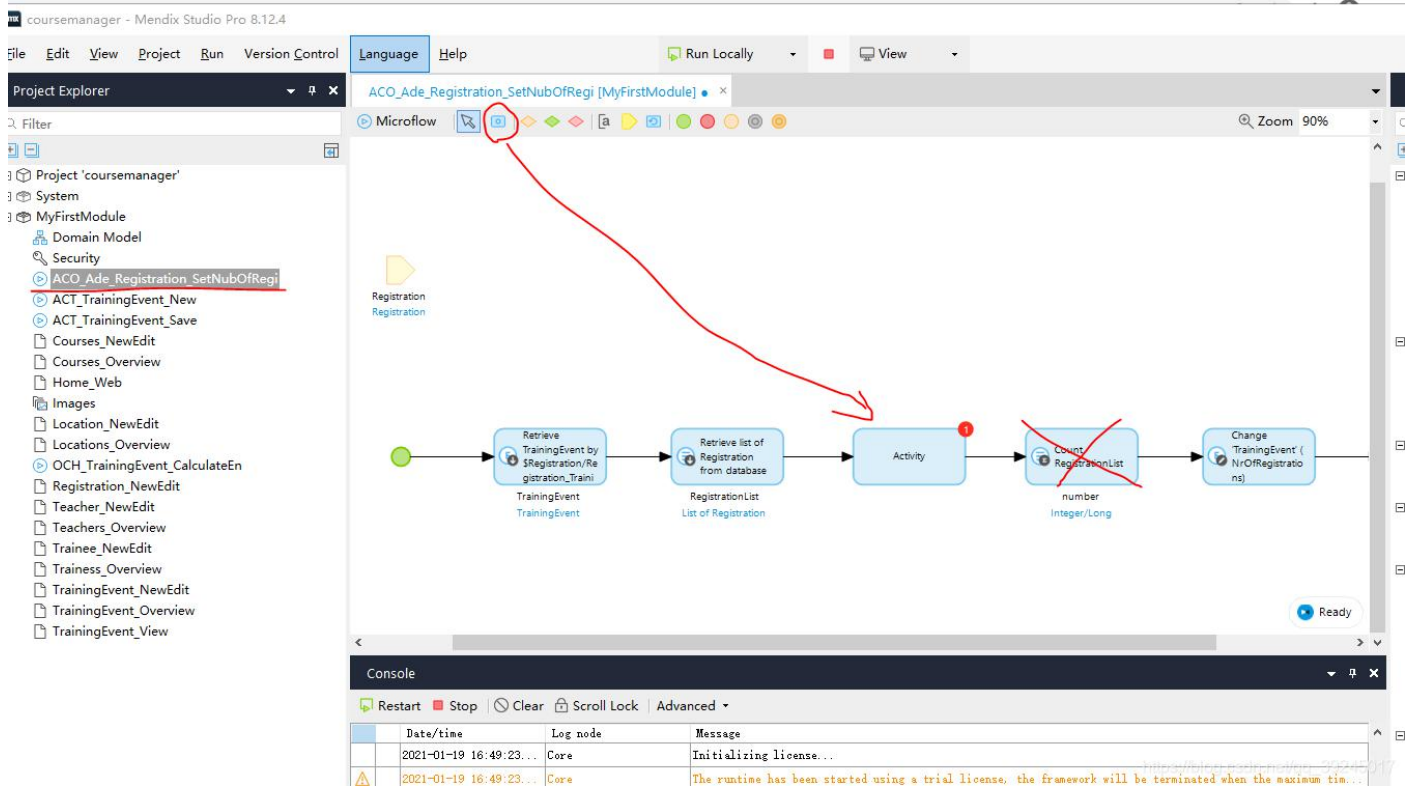
Password

[Sign in](#)

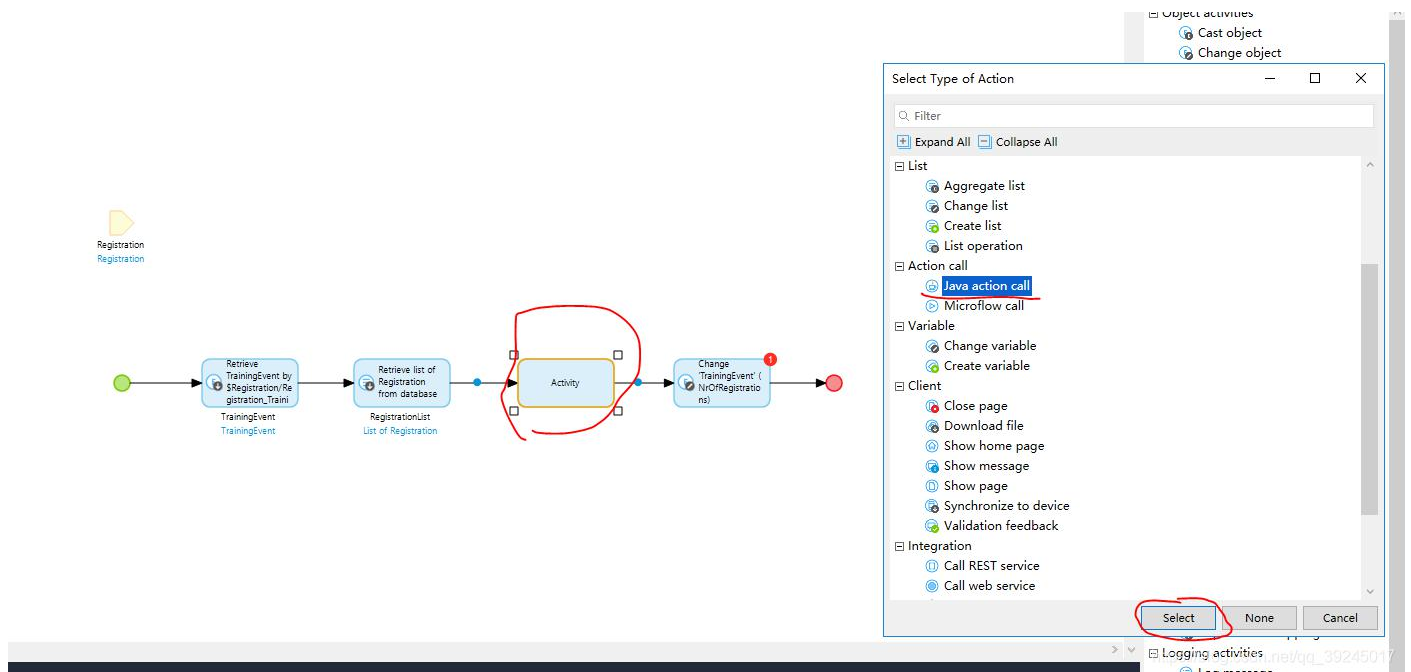
https://blog.csdn.net/qq_39245017

五、调用 JAVA 代码

1. 我们可以回到我们的项目中，双击我们之前创建的 ACO_Ade_Registration_SetNubOfRegi 用来数报名人数人数的微流，然后拖入一个 Activity 事件，将之前的 Count 事件删除。



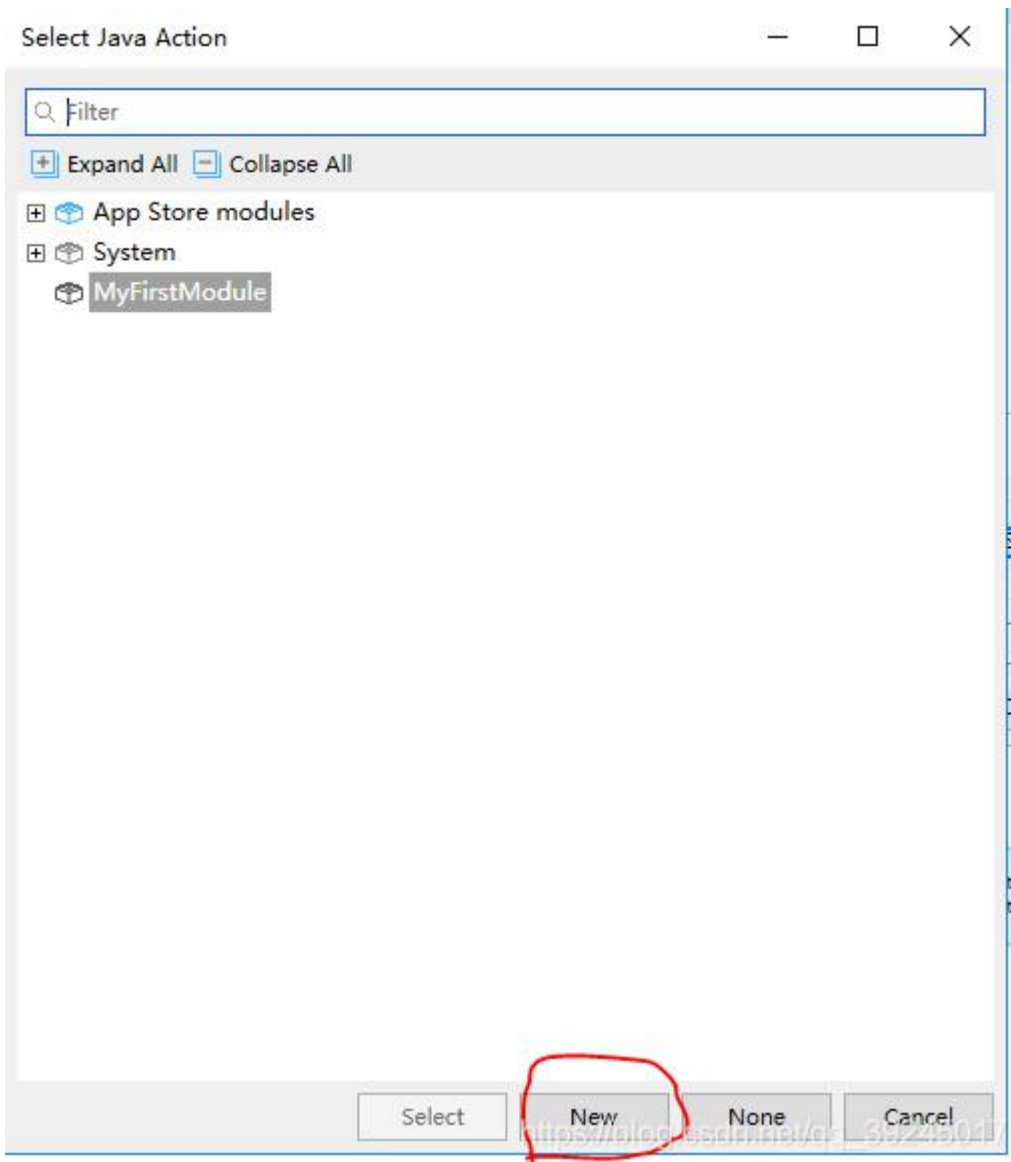
2. 删除以后，我们双击刚才拖入的事件，然后选择 Java action call 然后点击 Select 按钮。



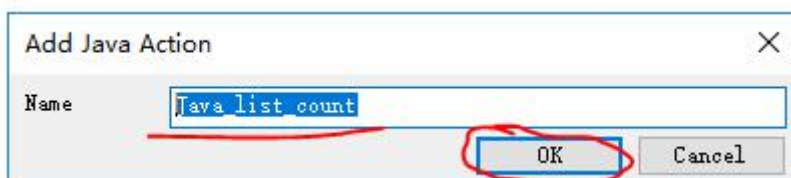
3. 我们选择一个 Java action。点击 Select 按钮



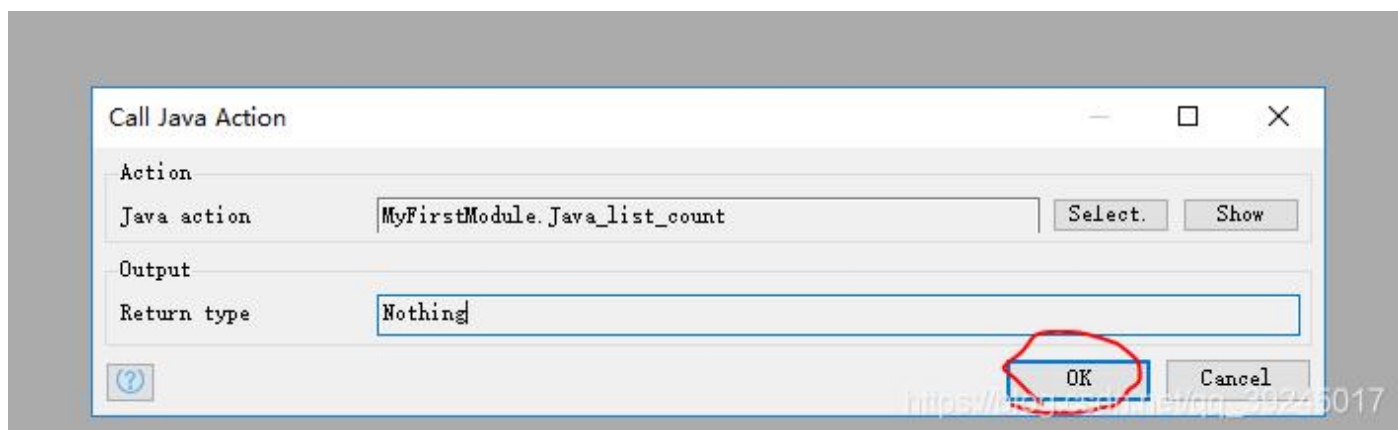
4. 这时候我们发现没有 javaAction，所以我们重新创建一个点击 New 按钮。



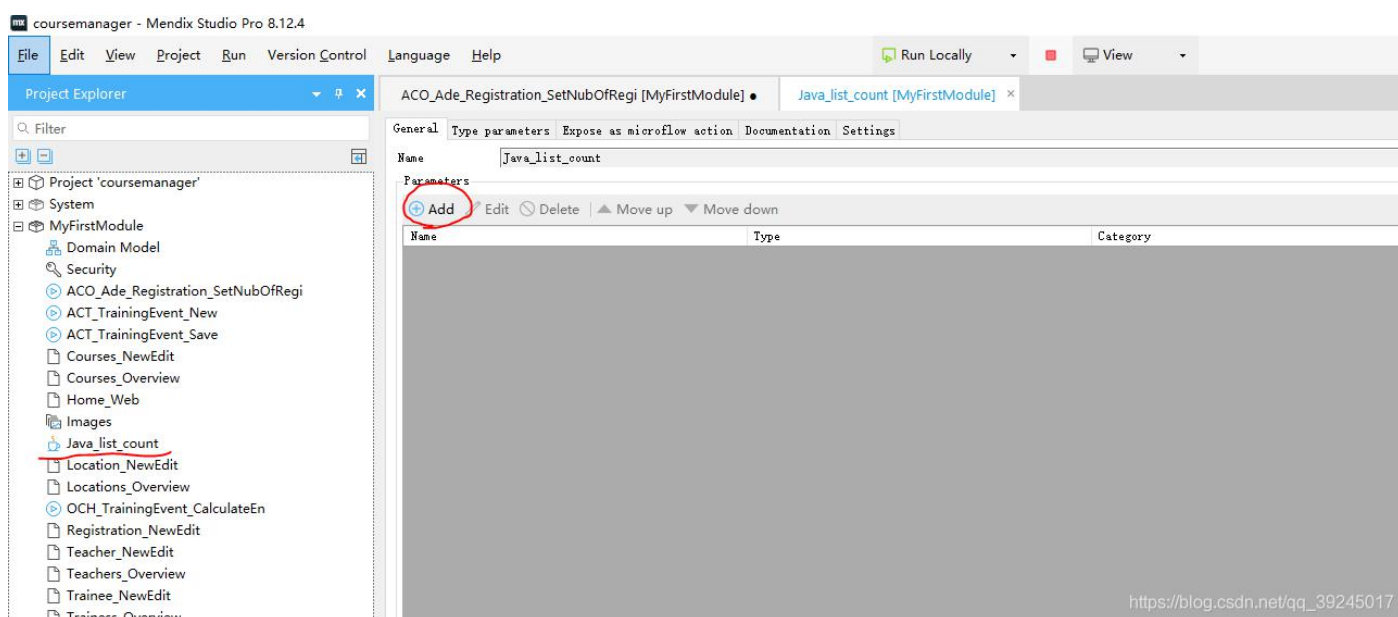
5. 在弹出的页面输入 java Action 名称为 Java_list_count 然后点击 Ok 按钮。



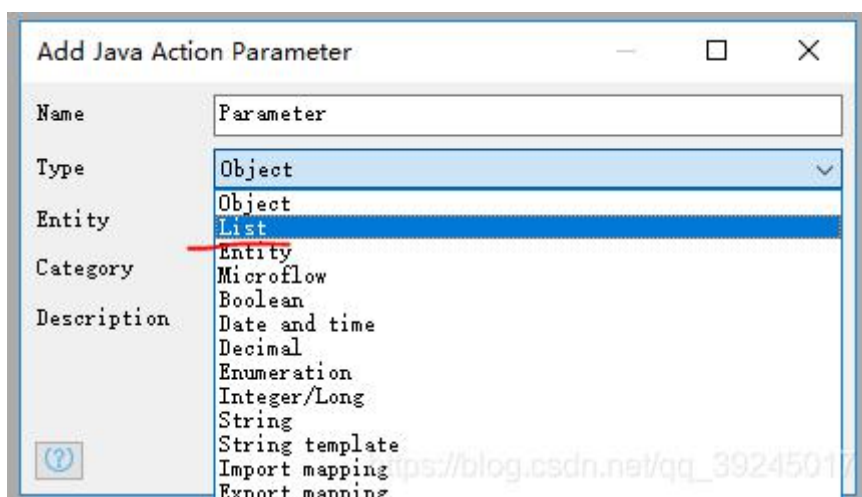
6. 这时候开发工具会跳转到我们刚才创建的那个 java action，我们只需要点击 Ok 按钮。



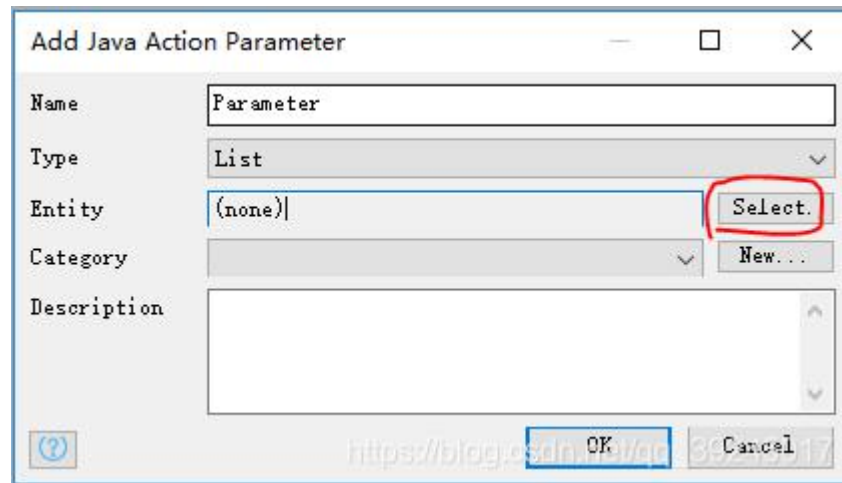
7. 然后我们需要添加参数，方便在 java 中处理。点击 Add 按钮。



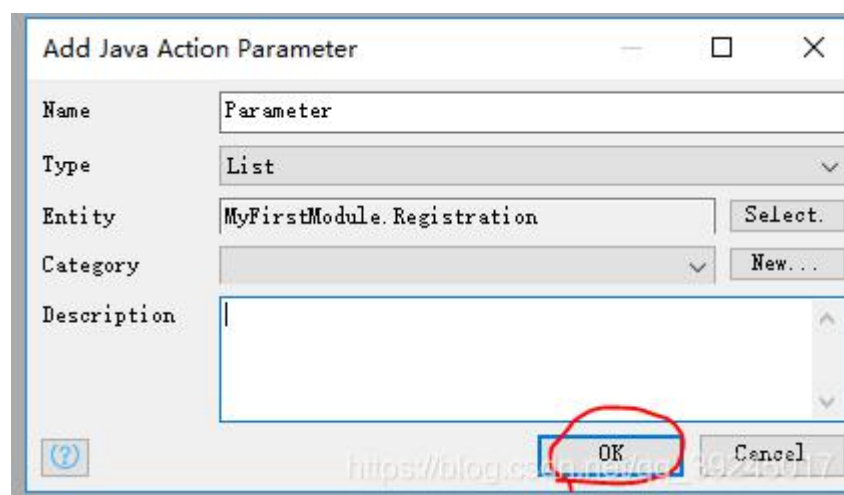
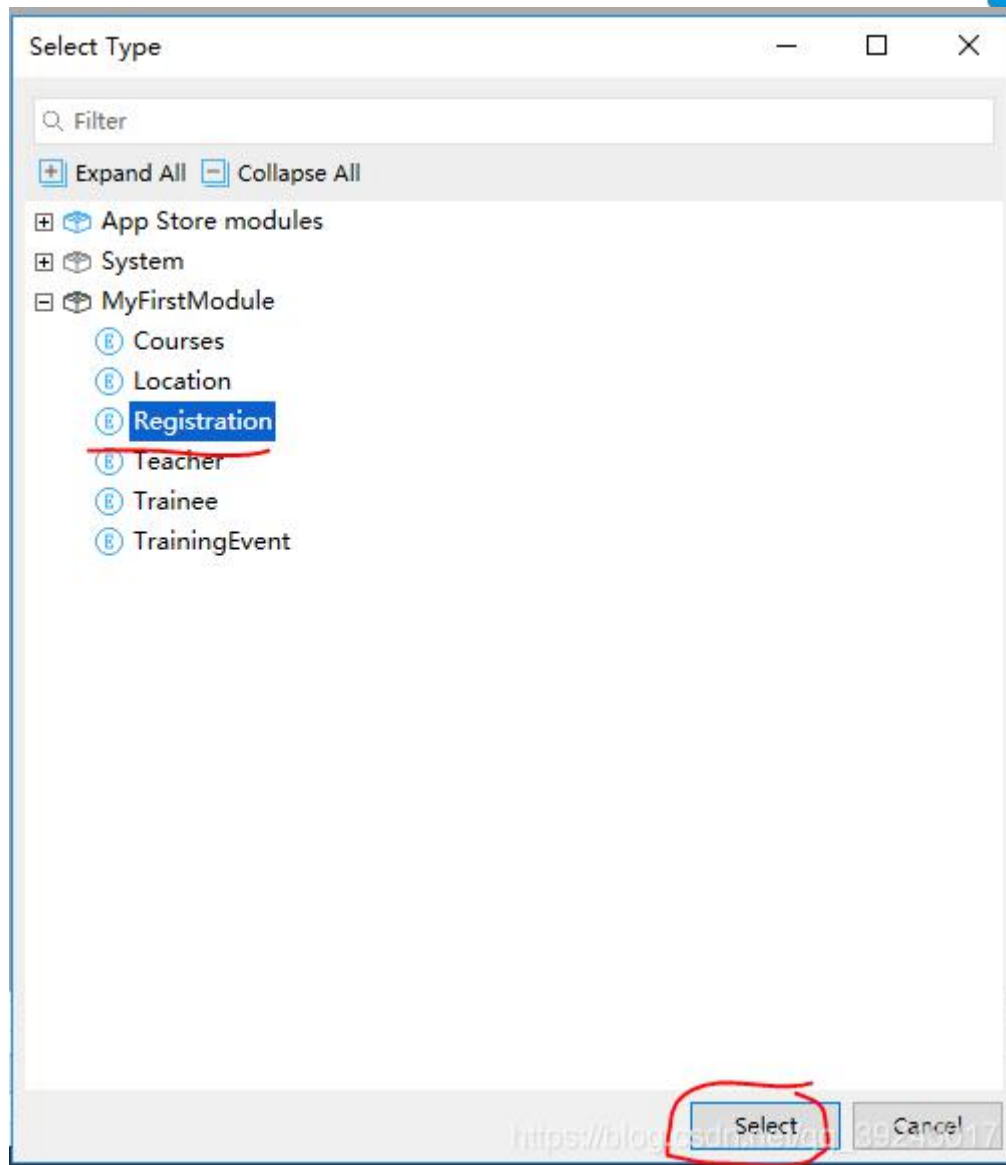
8. 在弹出的页面我们在 Type 中选择 List。



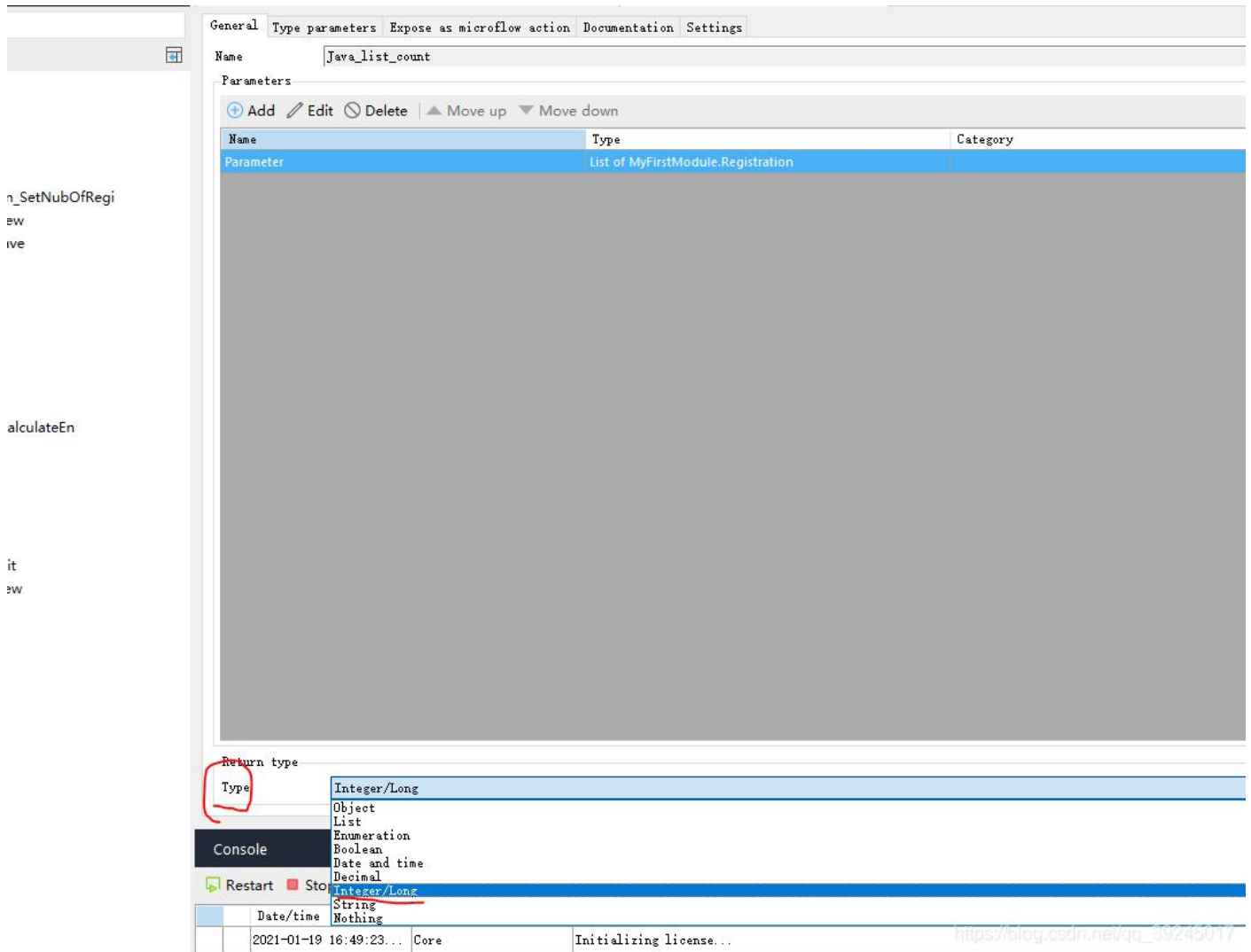
9. 然后 Select 选择我们需要数的数据模型。



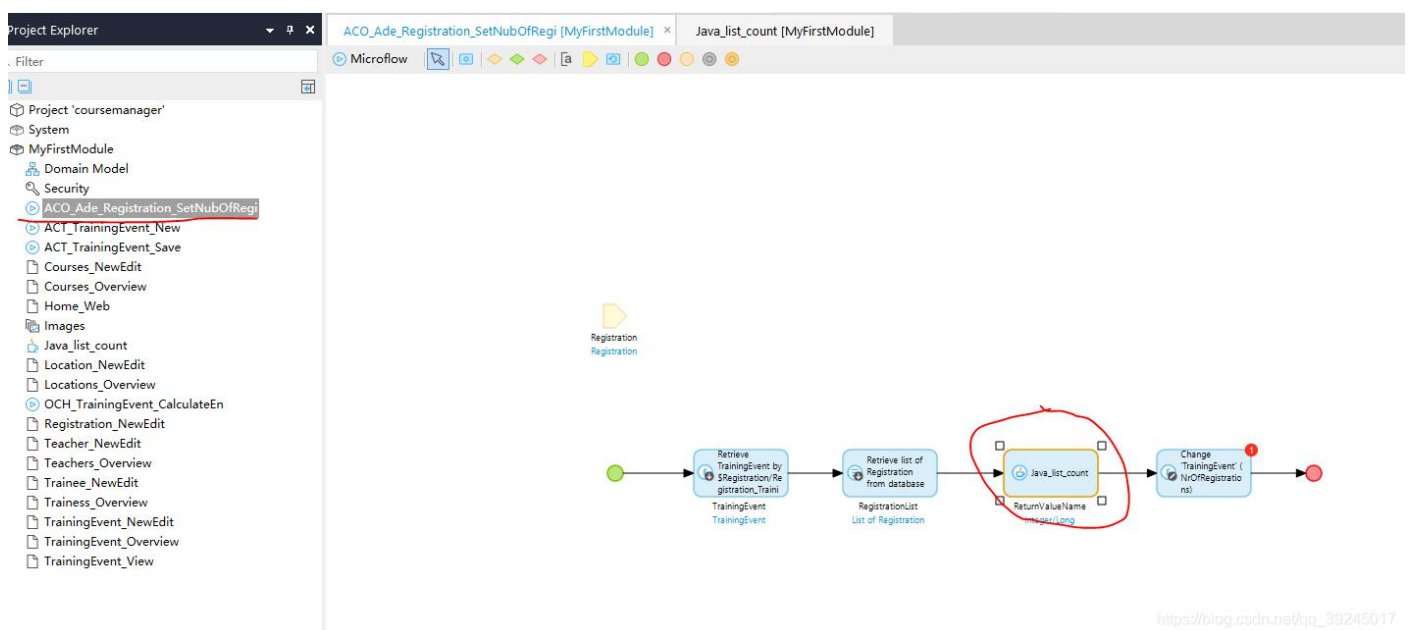
10. 然后选择 Registration(报名人员)数据模型，然后点击 Select->Ok 按钮。



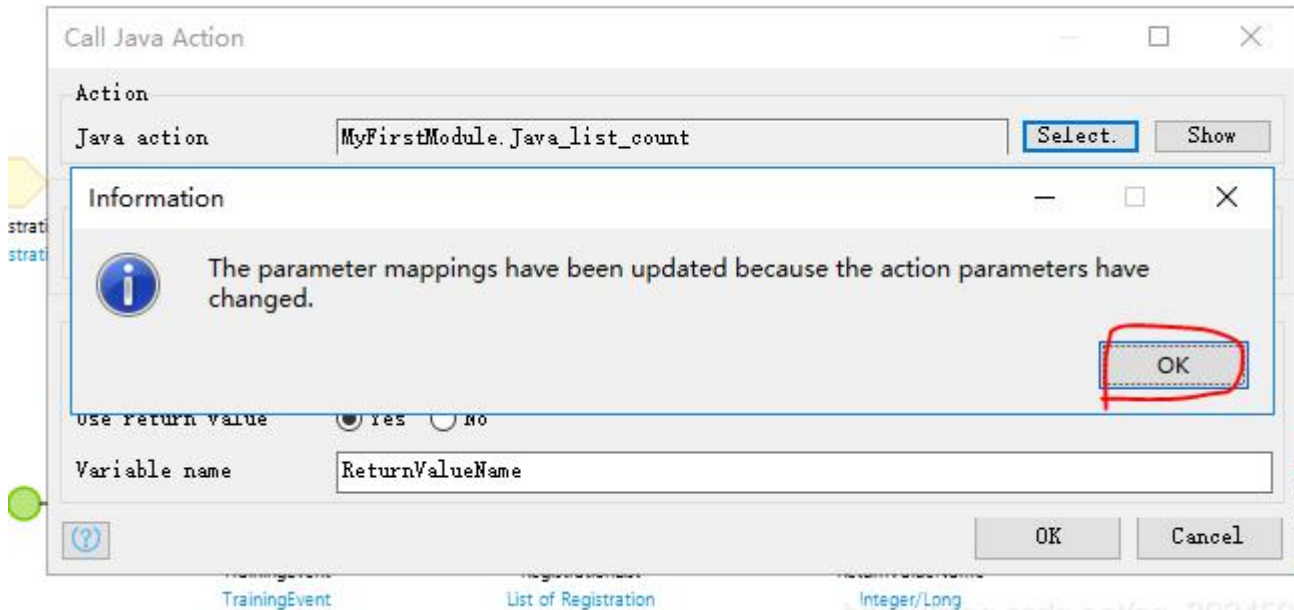
11. 然后我们需要返回一个参数，我们定义返回的参数为 Integer/Long 类型的数据。



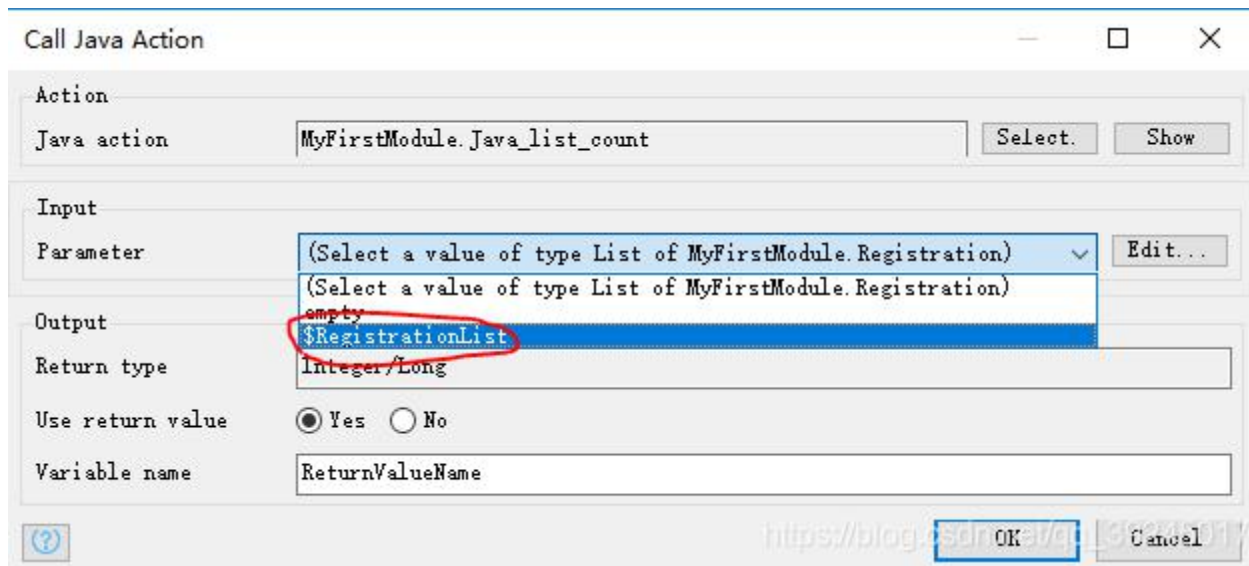
12. 设置好以后回到我们的微流，双击 java action。



13. 这时候会报错，我们不用管点击 Ok 按钮即可(说的是我们没有设置传入的参数)。



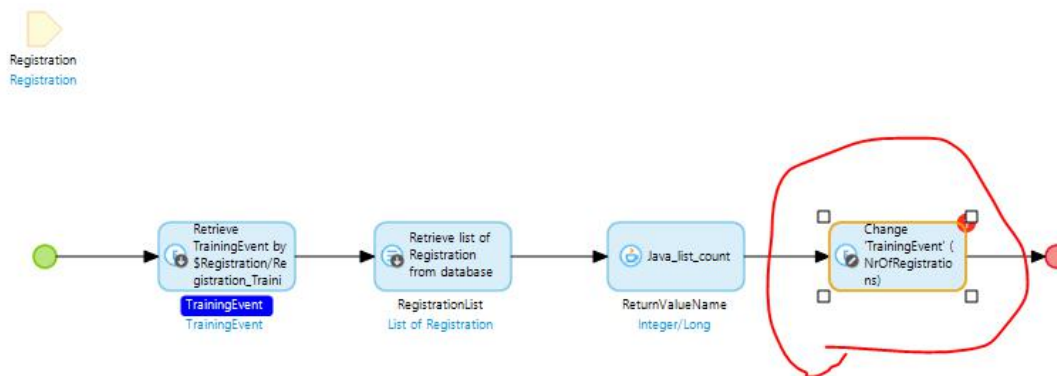
14. 我们设置传入的参数为\$RegistrationList。



15. 然后将返回的值命名为 listCount，然后点击 Ok 按钮。完成设置。



16. 然后我们双击最后的报错节点。



17. 然后双击我们的规则，将之前传入的参数 Znumber 改为 *number* 改为 listCount，然后点击 Ok->Ok 按钮。

Change Object

Input
Object: TrainingEvent (MyFirstModule.TrainingEvent)

Action
Commit: ☐ Yes ☐ Yes without events ☒ No
Refresh in client: ☐ Yes ☒ No

|

Member	Member type	Type	Value
NrOfRegistrations	Integer	Set	\$number

Edit Change Item

Member: NrOfRegistrations (Integer)

Type: ☒ Set ☐ Add ☐ Remove

Value: \$listCount

Line	Column	Error
------	--------	-------

Change Object

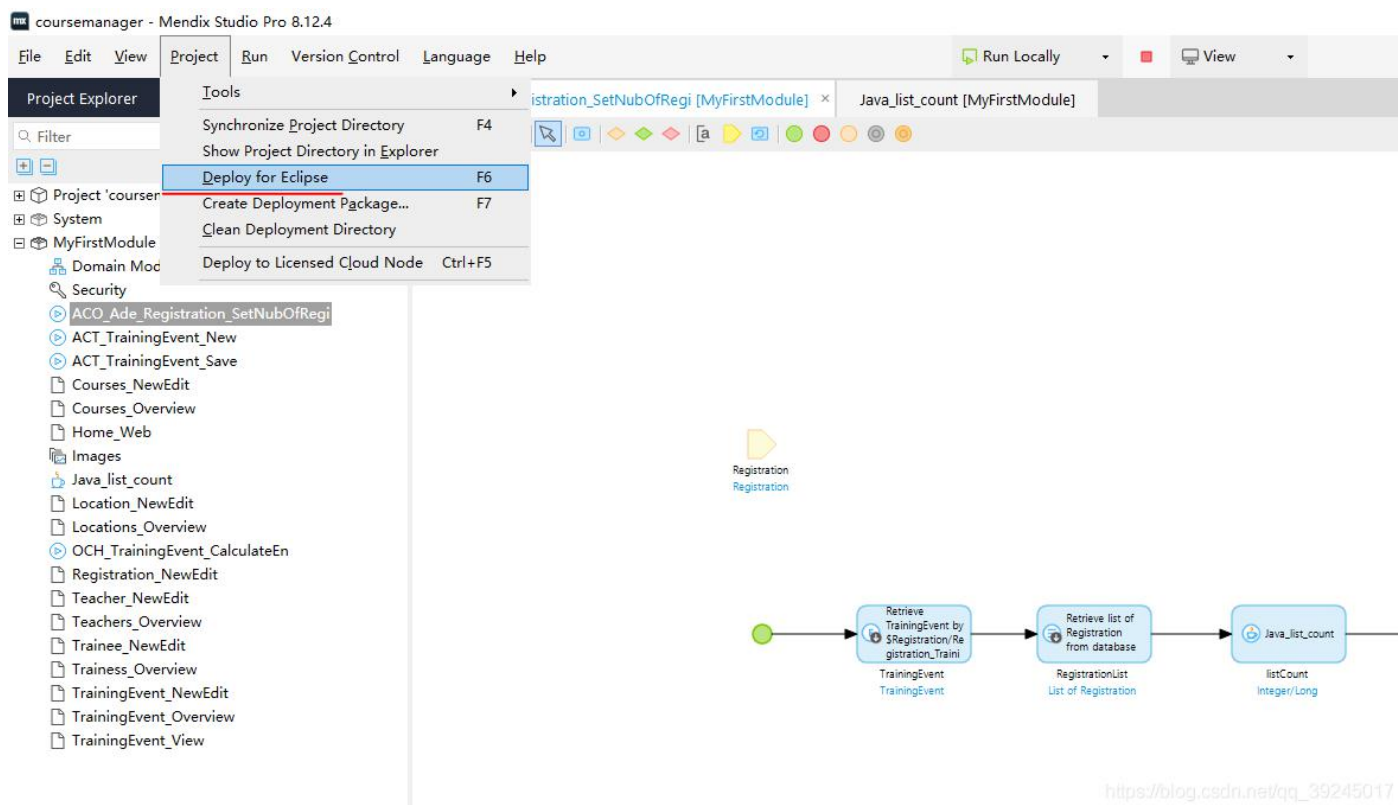
Input
Object: TrainingEvent (MyFirstModule.TrainingEvent)

Action
Commit: ☐ Yes ☐ Yes without events ☒ No
Refresh in client: ☐ Yes ☒ No

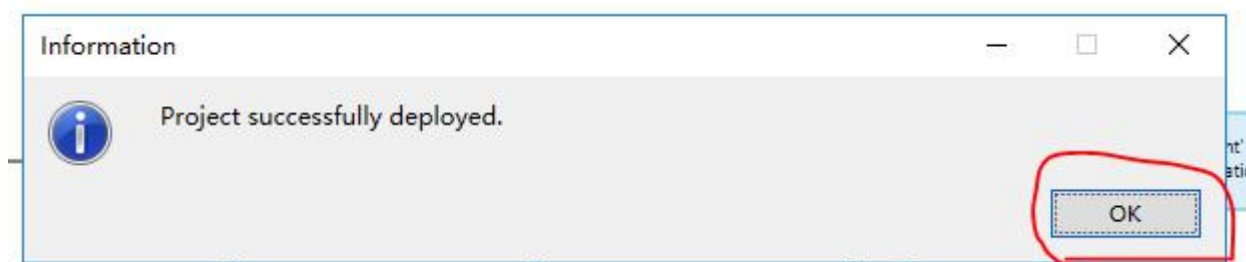
|

Member	Member type	Type	Value
NrOfRegistrations	Integer	Set	\$listCount

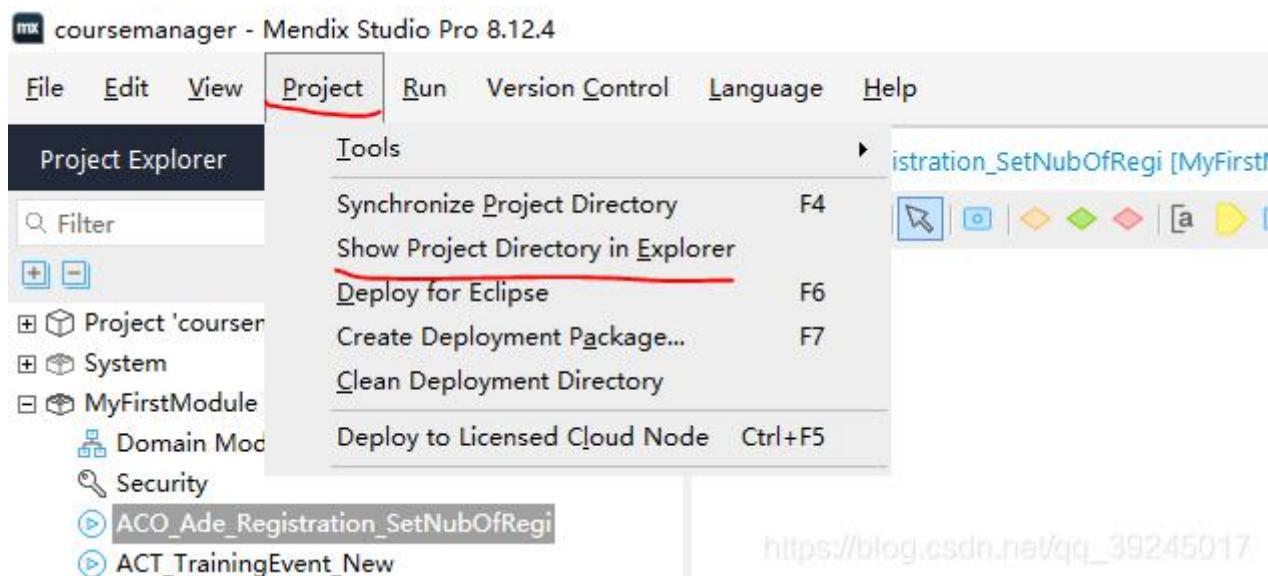
18. 然后我们保存一下项目，点击菜单栏的 Project 中的 Deploy for Eclipse 来生成 JAVA 代码。



19. 提示下列消息证明 Java 代码生成完毕，点击 Ok 按钮。



20. 生成完毕以后我们去找一下我们对应的代码，点击 Project 菜单找到 Show project Directory in Explorer(在电脑目录中打开我们的项目文件夹)



21. 在弹出的目录窗口我们可以看到 javasource，我们打开 javasource 找到我们的模块 myfirstmodule 然

后打开 actions 目录(该目录存放我们所有的 javaAction)，可以看到系统给我们创建的 java 类

Java_list_count.java

) > Mendix > shaoqunchao

名称	修改日期	类型	大小
deployment	2021/1/14 9:02	文件夹	
javascriptsource	2021/1/13 17:58	文件夹	
javasource	2021/1/14 9:02	文件夹	
resources	2021/1/13 17:58	文件夹	
theme	2021/1/13 17:58	文件夹	
userlib	2021/1/13 17:58	文件夹	
widgets	2021/1/13 17:58	文件夹	
.classpath	2021/1/19 17:29	CLASSPATH 文件	1 KB
.project	2021/1/19 17:29	PROJECT 文件	1 KB
coursemanager.mpr	2021/1/19 17:15	MPR 文件	13,724 KB
coursemanager.mpr.bak	2021/1/13 17:58	BAK 文件	13,472 KB
coursemanager.mpr.lock	2021/1/18 14:49	LOCK 文件	0 KB
shaoqunchao.launch	2021/1/19 17:29	LAUNCH 文件	4 KB

https://blog.csdn.net/qq_39245017

组织	新建	打开	选择
----	----	----	----

卷 (E:) > Mendix > shaoqunchao > javasource

名称	修改日期	类型	大小
administration	2021/1/14 9:02	文件夹	
atlas_ui_resources	2021/1/14 9:02	文件夹	
myfirstmodule	2021/1/19 17:29	文件夹	
system	2021/1/19 17:29	文件夹	

https://blog.csdn.net/qq_39245017

卷 (E:) > Mendix > shaoqunchao > javasource > myfirstmodule

名称	修改日期	类型	大小
actions	2021/1/19 17:29	文件夹	
datasets	2021/1/14 9:02	文件夹	
proxies	2021/1/15 17:39	文件夹	

https://blog.csdn.net/qq_39245017

卷 (E:) > Mendix > shaoqunchao > javasource > myfirstmodule > actions

名称	修改日期	类型	大小
Java_list_count.java	2021/1/19 17:29	IntelliJ IDEA	2 KB

https://blog.csdn.net/qq_39245017

22. 然后我们右击用 Idea 打开(用其他工具打开也可以)，我们在注释的地方写我们的代码。

```

    super(context);
    this.__Parameter = Parameter;
}

@Override
public java.lang.Long executeAction() throws Exception
{
    this.Parameter = new java.util.ArrayList<myfirstmodule.proxies.Registration>();
    if (__Parameter != null)
        for (IMendixObject __ParameterElement : __Parameter)
            this.Parameter.add(myfirstmodule.proxies.Registration.initialize(getContext(), __ParameterElement));

    // BEGIN USER CODE
    throw new com.mendix.systemwideinterfaces.MendixRuntimeException("Java action was not implemented");
    // END USER CODE
}

/**
 * Returns a string representation of this action
 */
@Override
public java.lang.String toString() { return "Java_list_count"; }

// BEGIN EXTRA CODE
// END EXTRA CODE

```

https://blog.csdn.net/qq_39245017

23. 输入以下代码，并保存。

24. `return Long.valueOf(__Parameter.size());`

```

    super(context);
    this.__Parameter = Parameter;
}

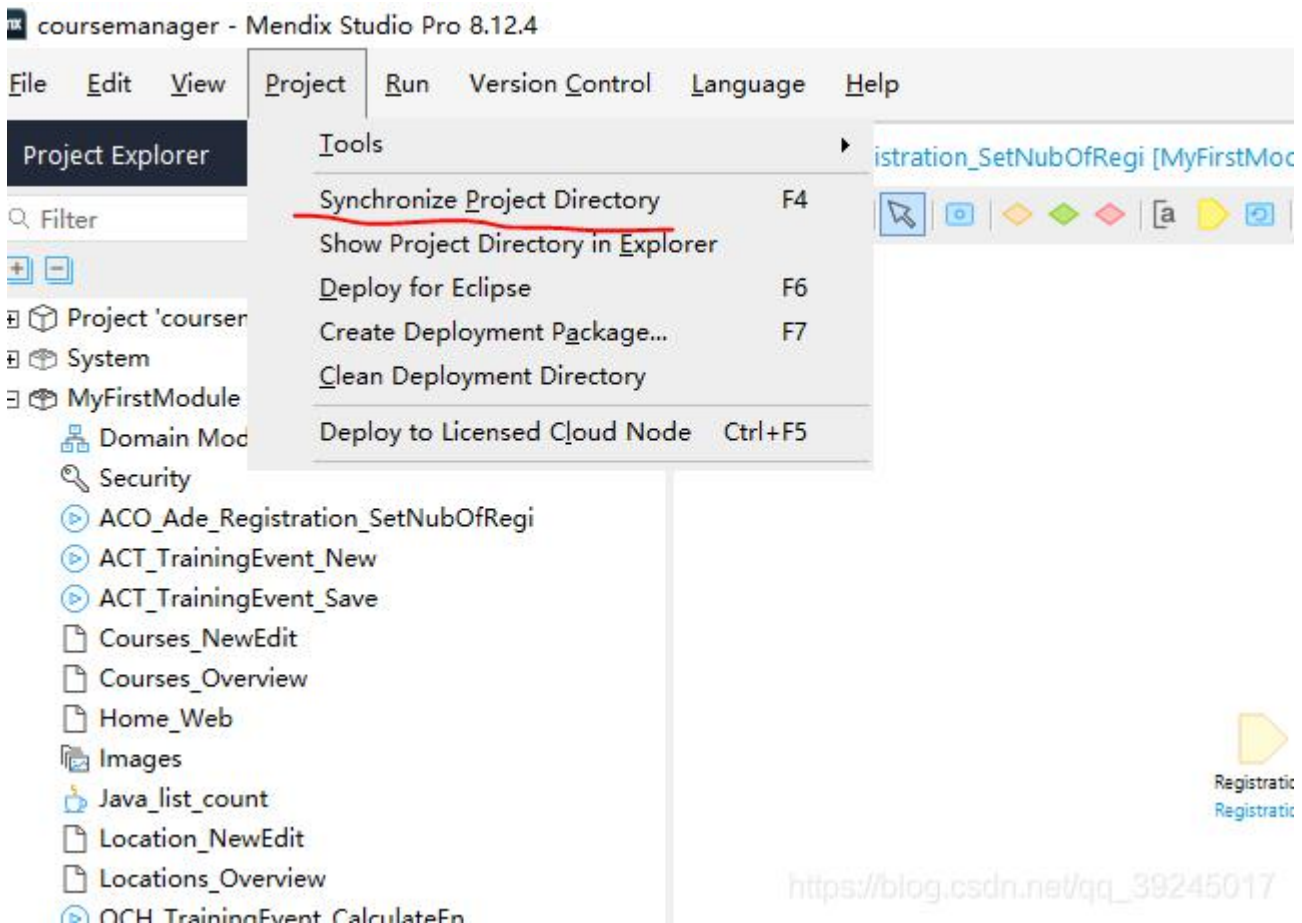
@Override
public java.lang.Long executeAction() throws Exception
{
    this.Parameter = new java.util.ArrayList<myfirstmodule.proxies.Registration>();
    if (__Parameter != null)
        for (IMendixObject __ParameterElement : __Parameter)
            this.Parameter.add(myfirstmodule.proxies.Registration.initialize(getContext(), __ParameterElement));

    // BEGIN USER CODE
    return Long.valueOf(__Parameter.size());
    // END USER CODE
}
/**

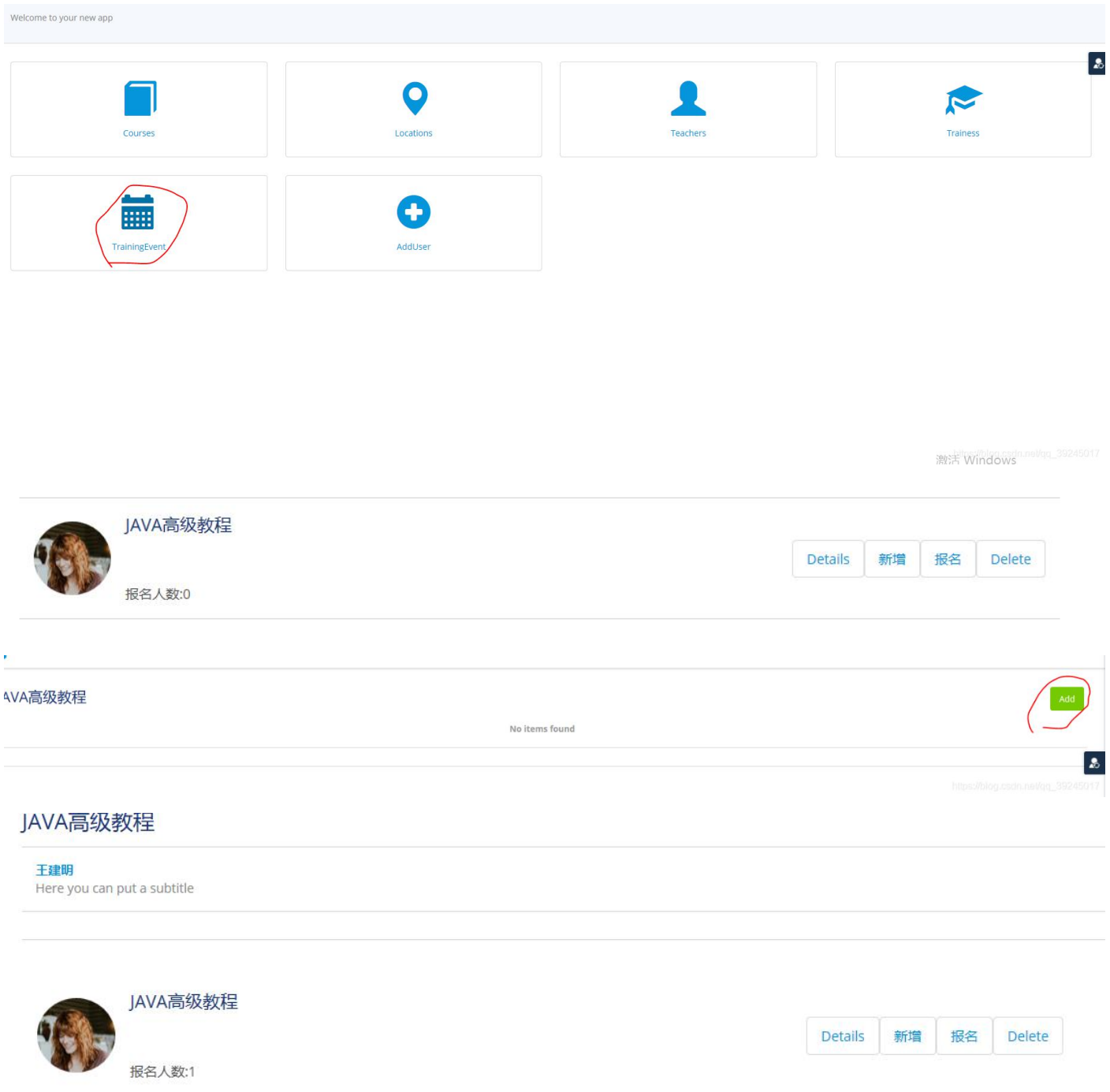
```

https://blog.csdn.net/qq_39245017

24. 然后我们回到 Medix 开发工具中，我们选择 project 菜单下的 Synchronize Project Directory(同步本地的文件)

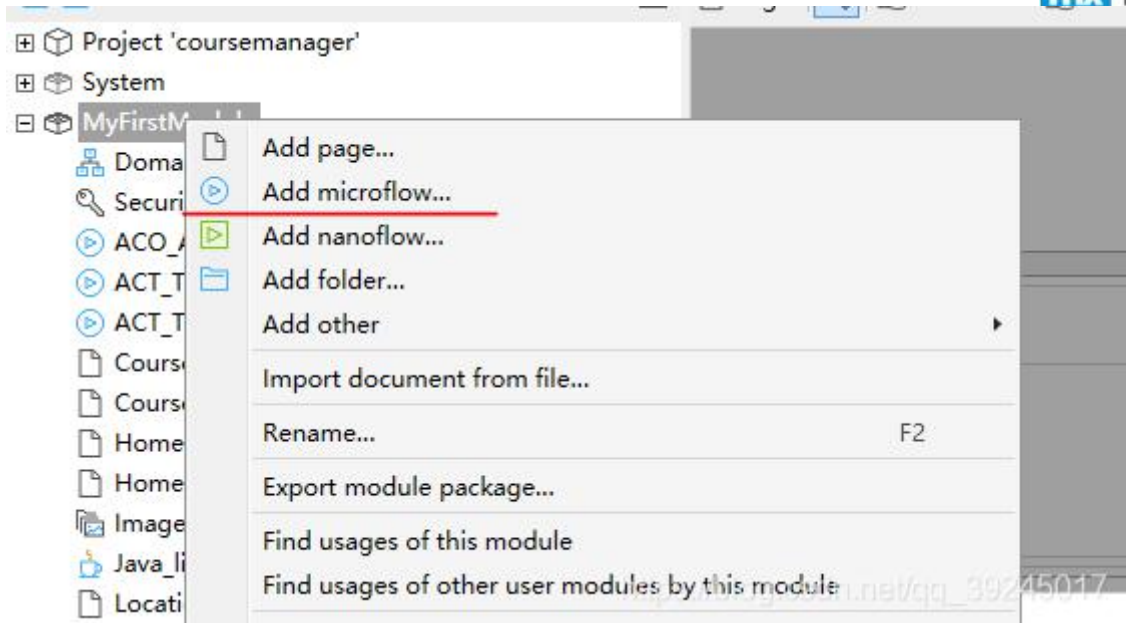


25. 然后我们运行项目进行测试。发现已经实现我们之前的效果了。该功能到此就结束了。

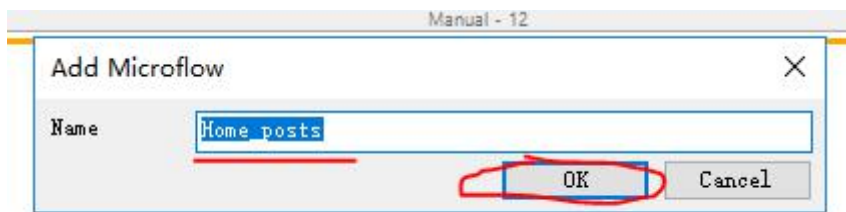


六、调用外部接口

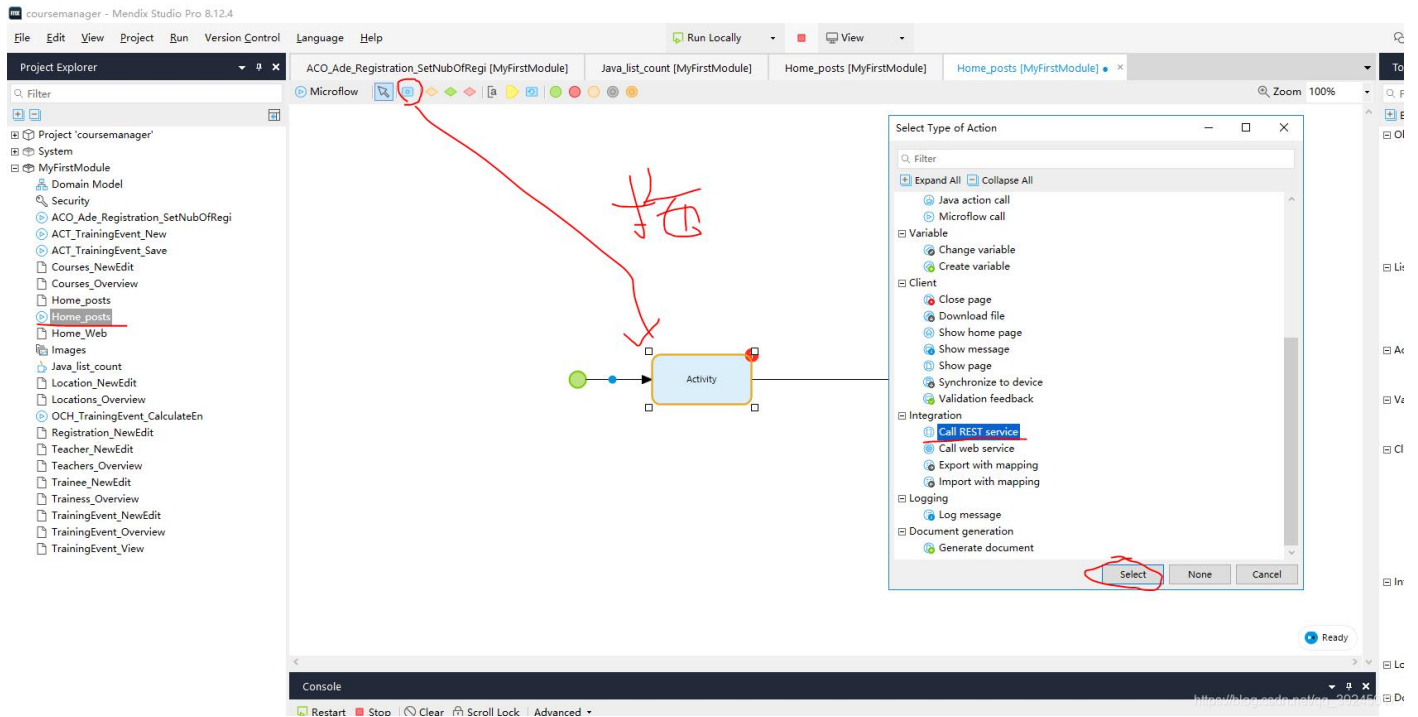
1. 首先我们需要创建一个 Microflow 微流。



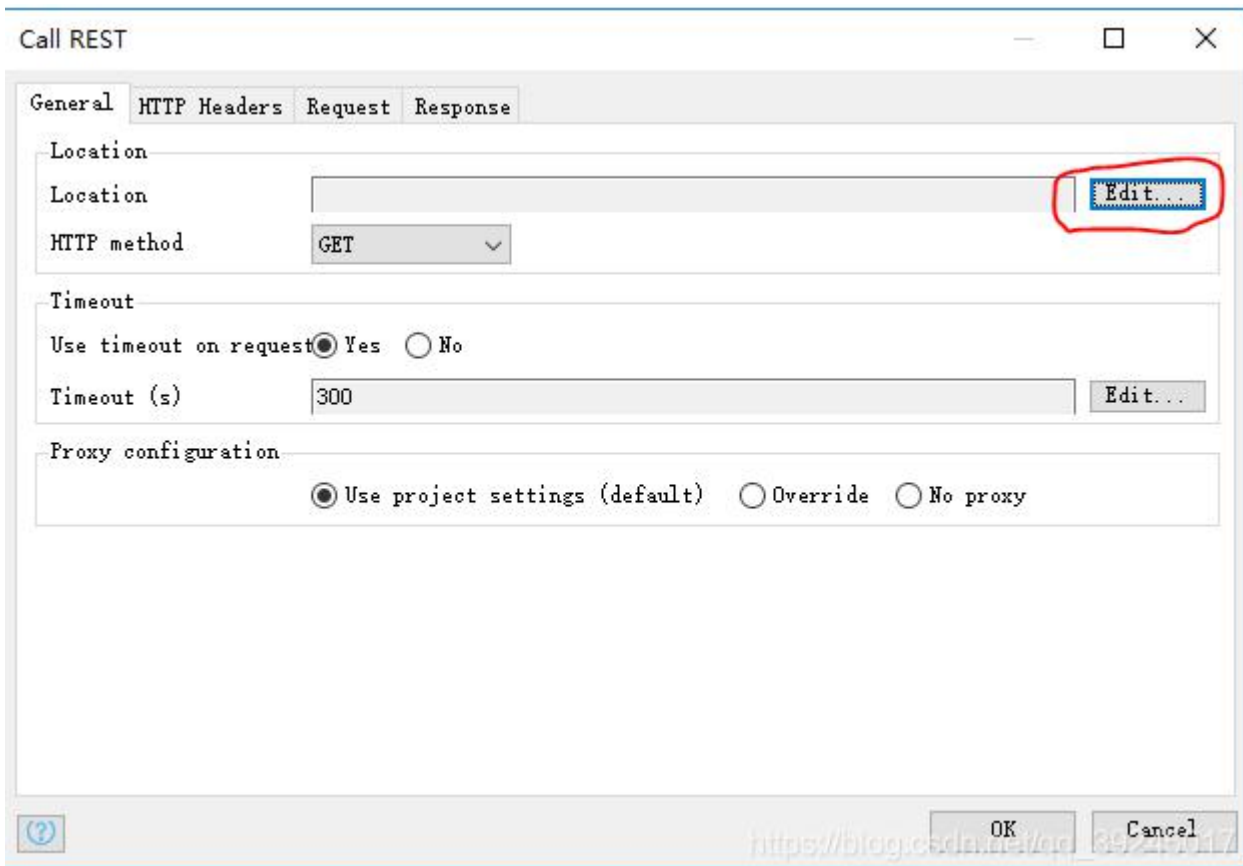
2. 在弹出窗口输入微流的名称 Home_posts。



3. 然后打开创建的微流，我们拖入一个 Activity 事件，然后双击选择 Call REST service，然后点击 Select 按钮

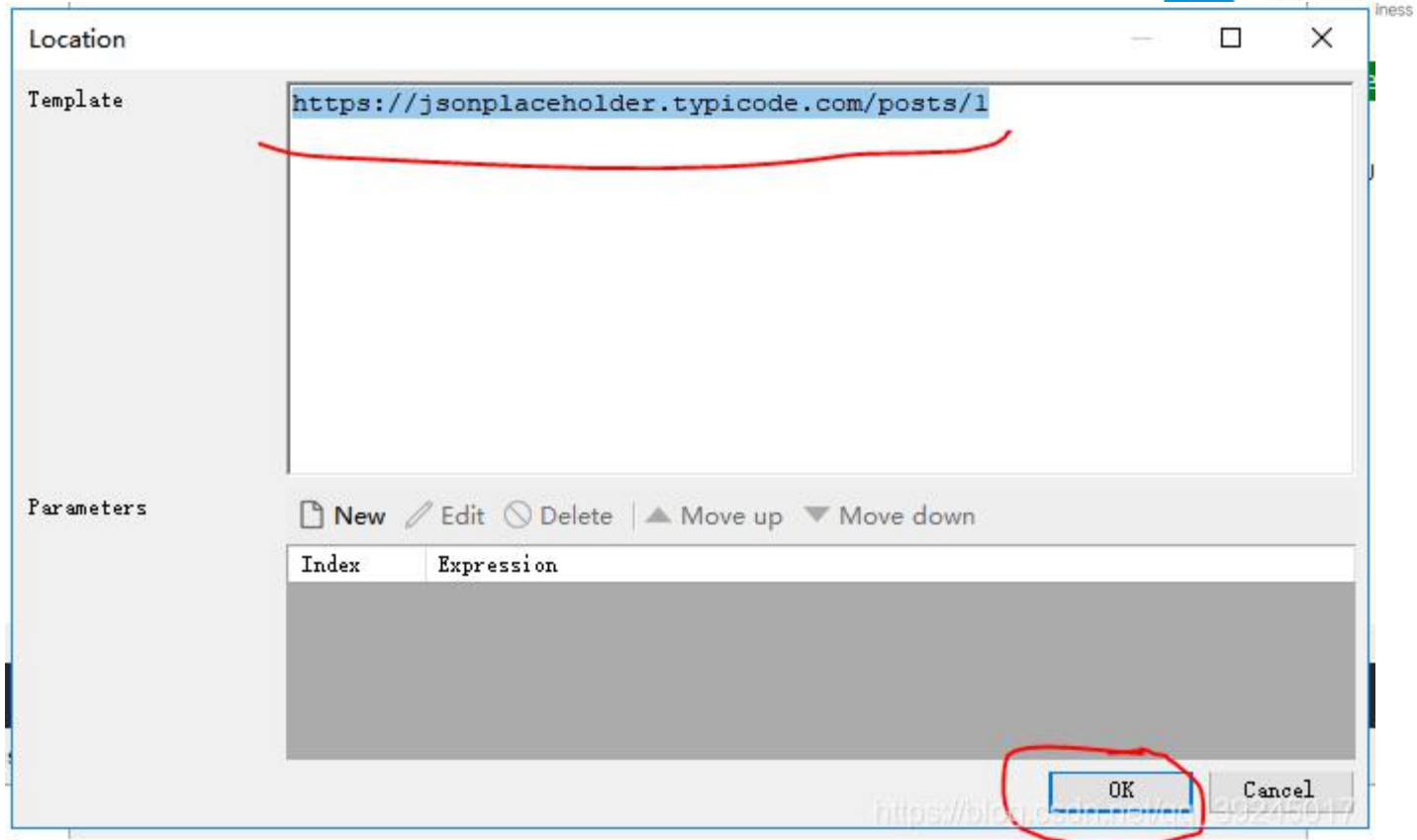


4. 在弹出的页面我们编辑调用的连接地址。



5. 输入连接地址，然后点击 OK 按钮。

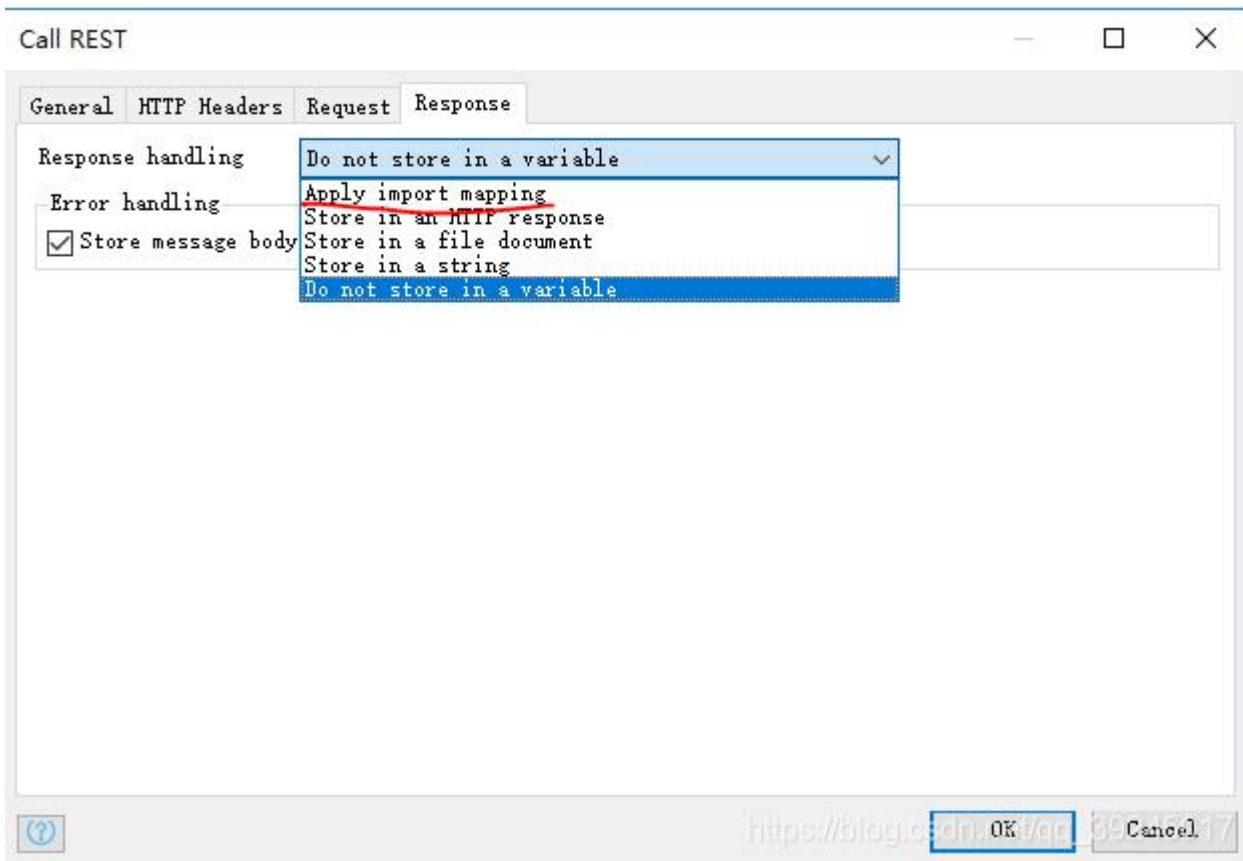
6. <https://jsonplaceholder.typicode.com/posts/1> 是一个外部的接口，用于测试 REST 接口调用



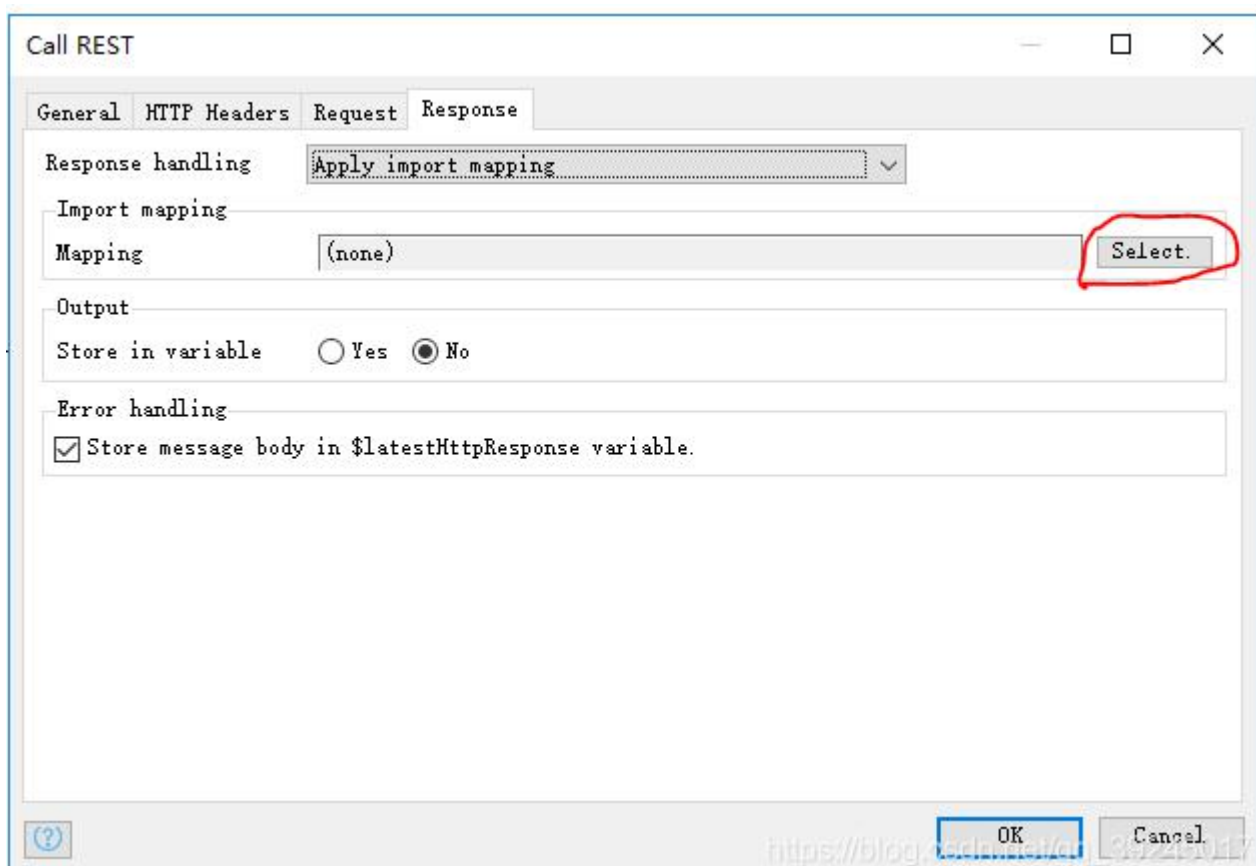
6. 然后其余的我们就不用操作，只处理 Response，我们在 Response handling 选择 Apply import mapping 方式。

- Do not store in a variable(只调用不处理)
- Store in a String(将调用结果存入一个字符串里)
- Store in a file document(将调用结果存在文件里)
- Store in an HTTP response(将调用结果存在 response 数据结构中)

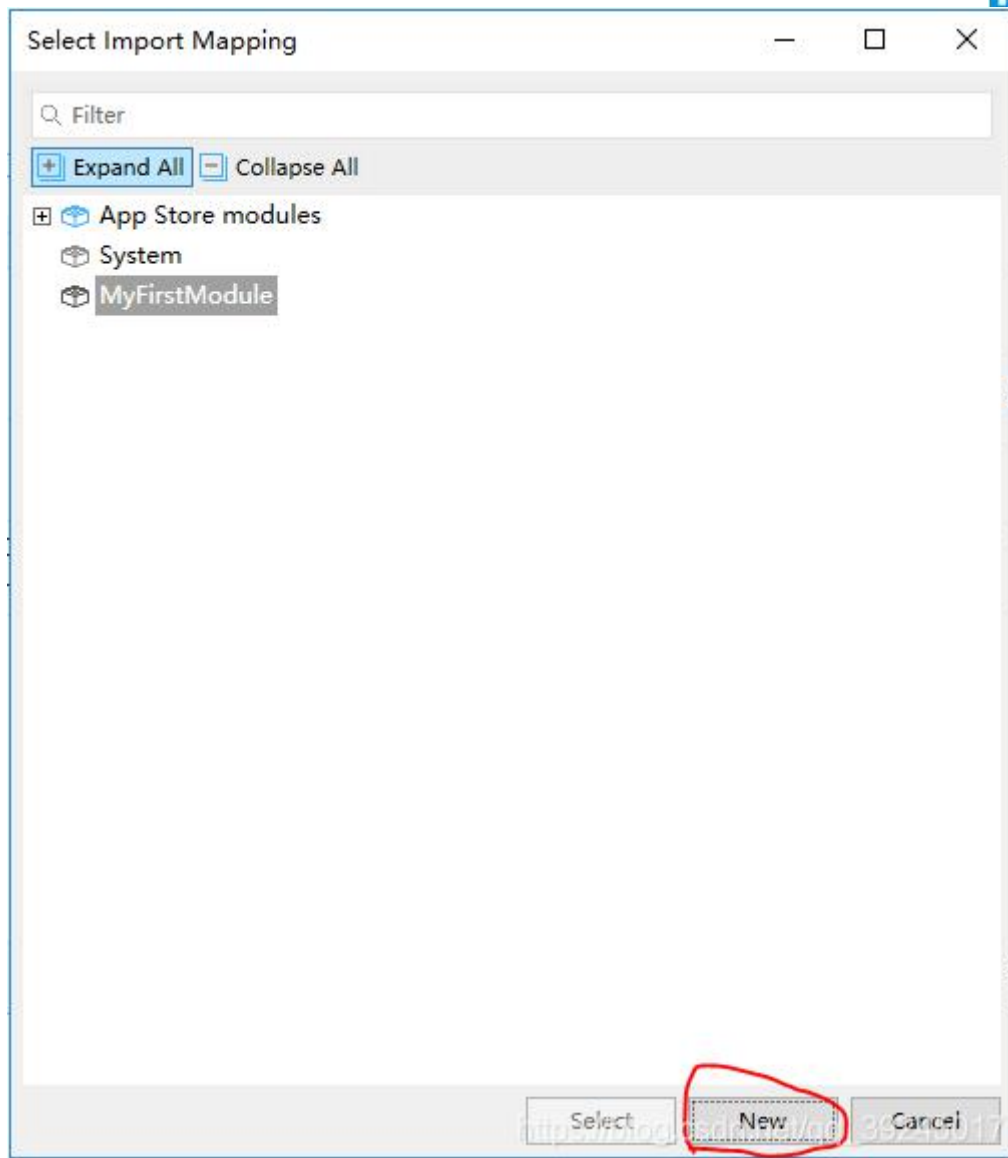
- Apply import mapping(将调用结果存入到内存中的 Json 对象)。我们选择存入到一个 json 中



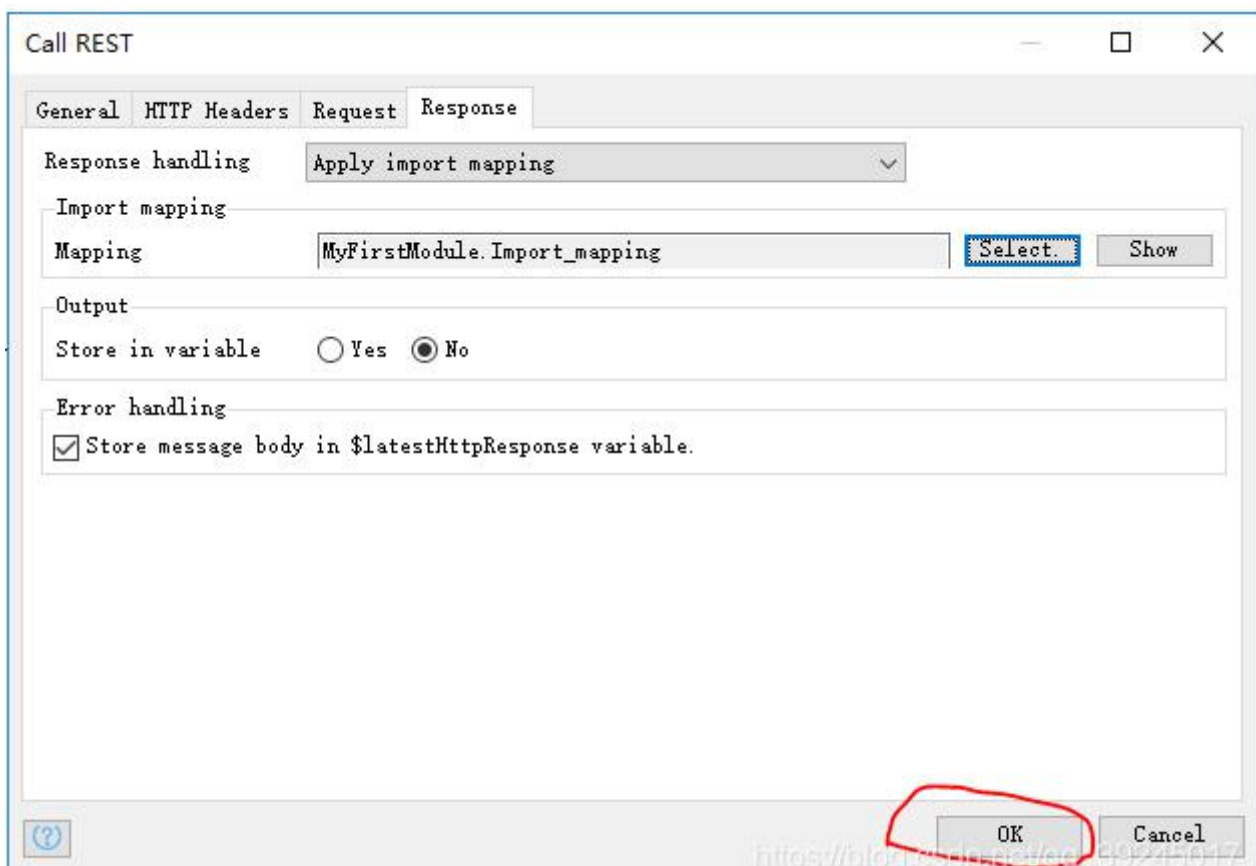
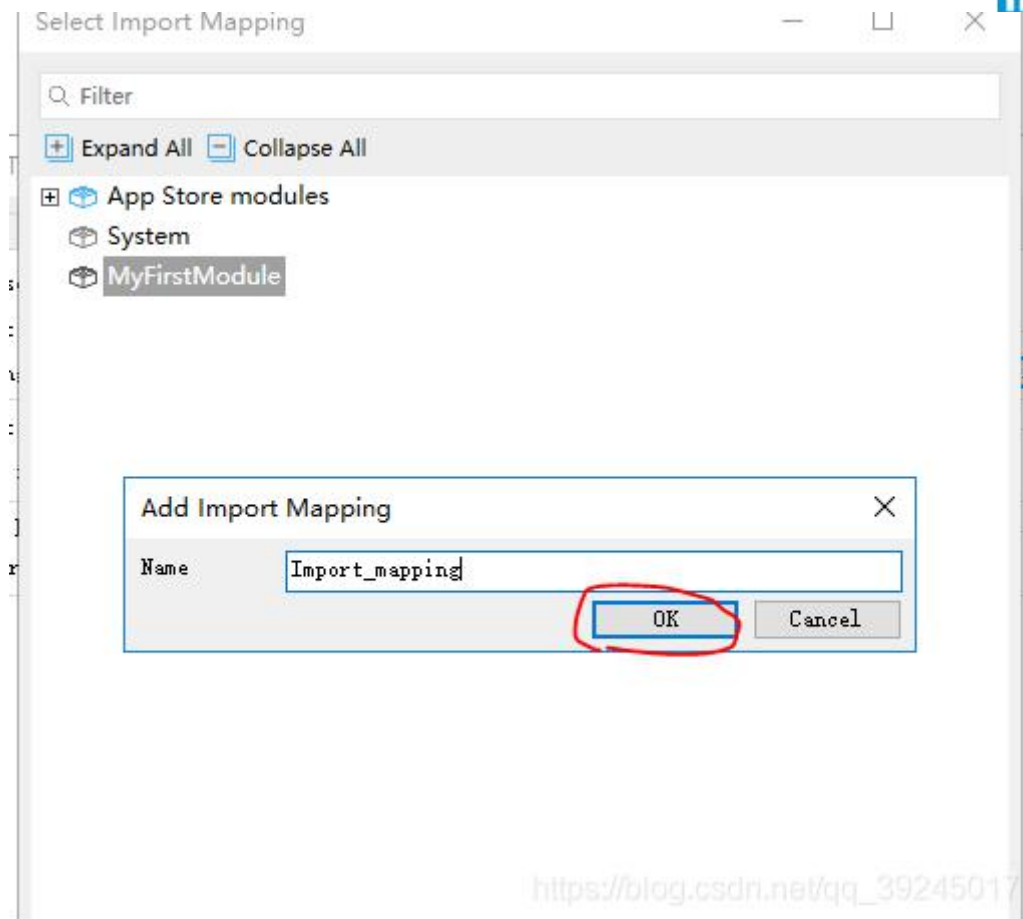
7. 然后我们需要先定义 mapping，点击 Select 按钮。



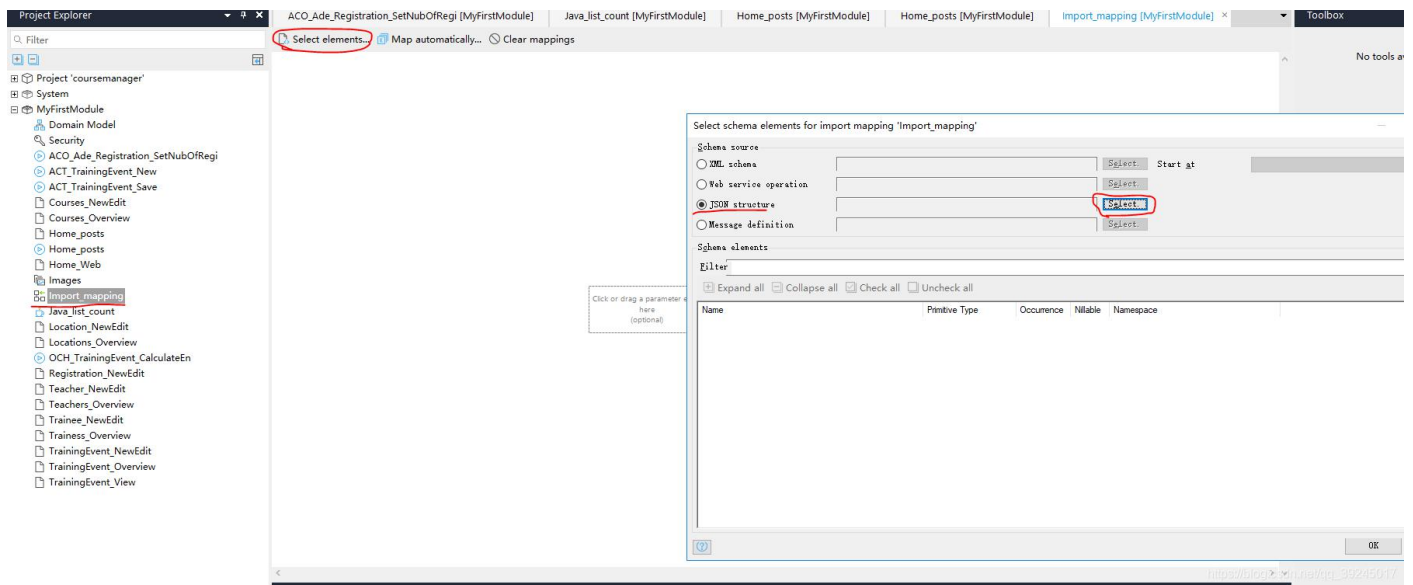
8. 点击 New 按钮新建。



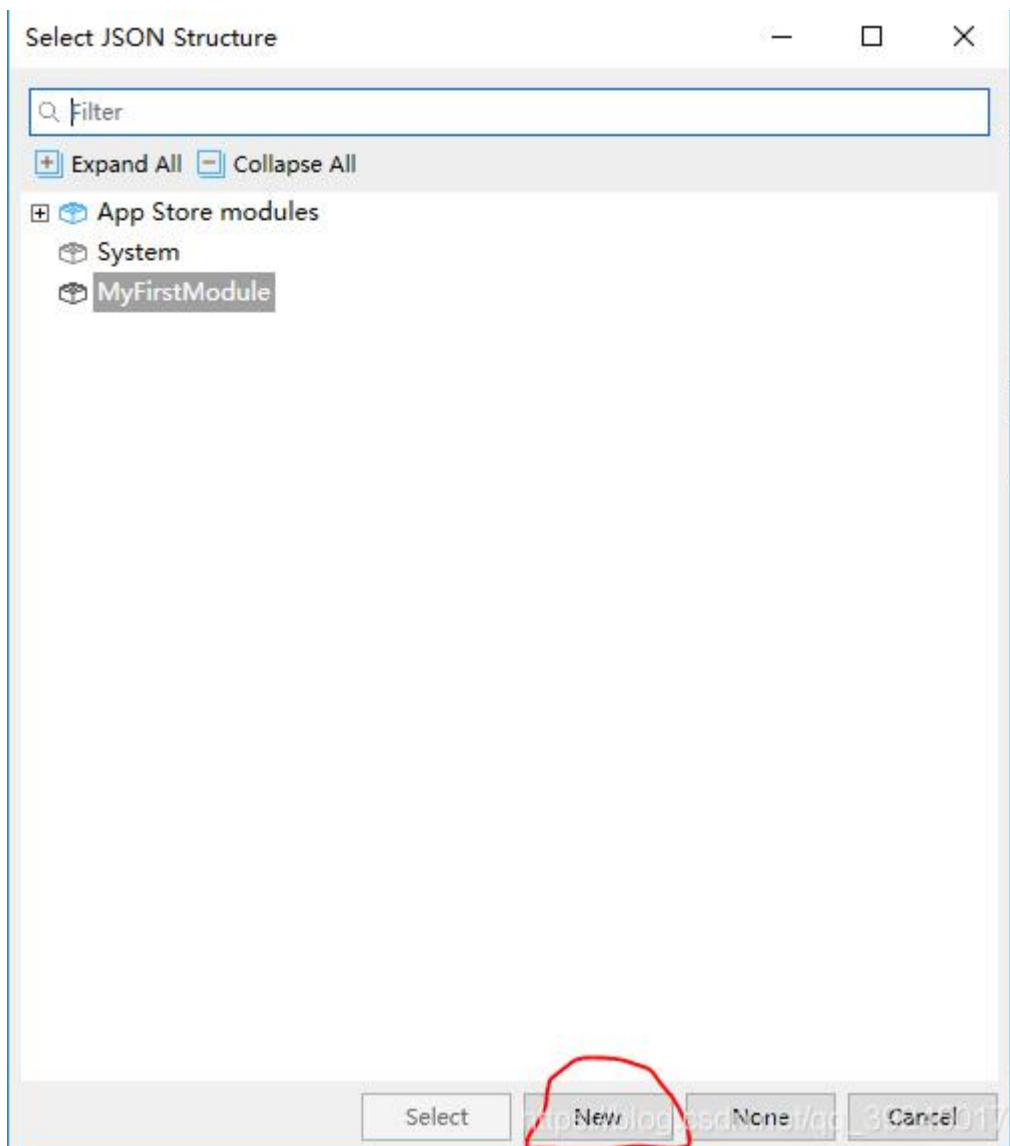
9. 然后直接点击 Ok->Ok 按钮。



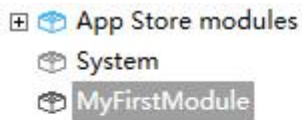
10. 然后我们在目录树中找到我们刚才创建的 mapping 然后双击，点击左上角的 select elements 选择我们的 mapping。



11. 我们点击 New 按钮新创建一个。



12. 然后点击 Ok 按钮。



https://blog.csdn.net/qq_39245017

13. 然后在弹出的页面输入请求过后的响应体，我这边就直接罗列在下方。可直接复制，复制好以后点击 Refresh 按钮刷新一下，Mendix 开发工具会自动的将响应体解析出来。然后点击 Ok 完成创建。

```
{  
  "userId": 1,  
  "id": 1,  
  "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",  
  "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"  
}
```

General	Documentation
---------	---------------

JSON snippet

Format

```
{  
    "userId": 1,  
    "id": 1,  
    "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",  
    "body": "quia et suscipit\nnsuscipit recusandae consequuntur expedita et cum  
\nreprehenderit molestiae ut ut quas \t\ttotam\nnostrum rerum est autem sunt rem eveniet  
architecto"  
}
```

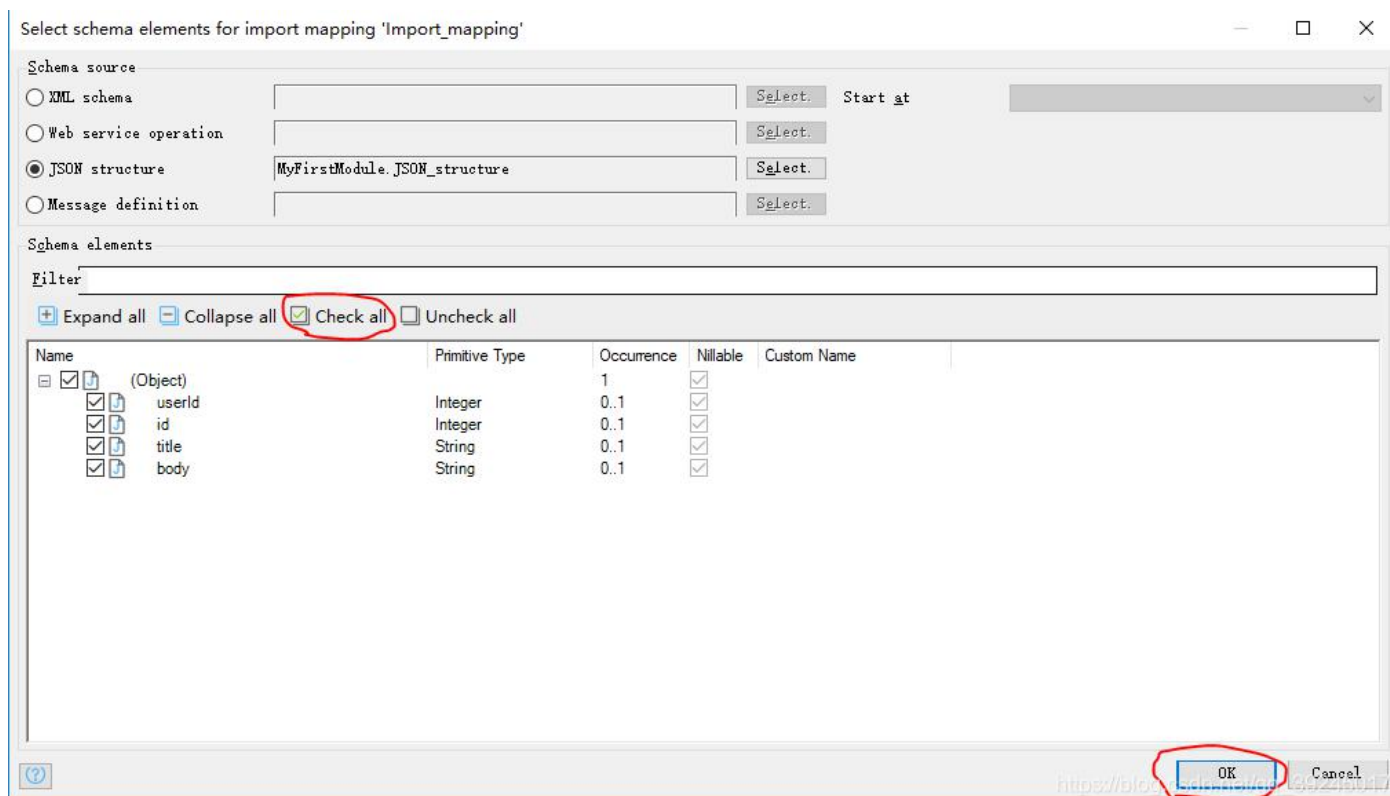
Structure

Filter

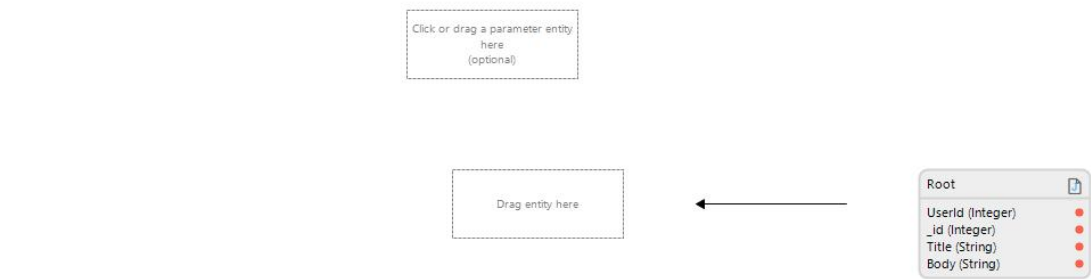
 Expand all  Collapse all  Refresh

Name	Value	Primitive Type	Occurrence	Nullable	Custom Name
(Object)			1	☒	Root
userId	1	Integer	0..1	☒	UserId
id	1	Integer	0..1	☒	_id
title	"sunt aut facere re..."	String	0..1	☒	Title
body	"quia et suscipitns..."	String	0..1	☒	Body

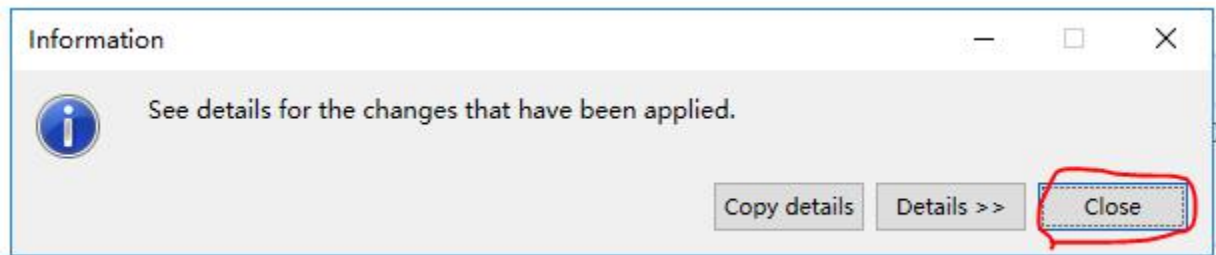
14. 我们需要将所有信息进行返回，点击 Check all 按钮选中所有，然后点击 Ok 按钮。



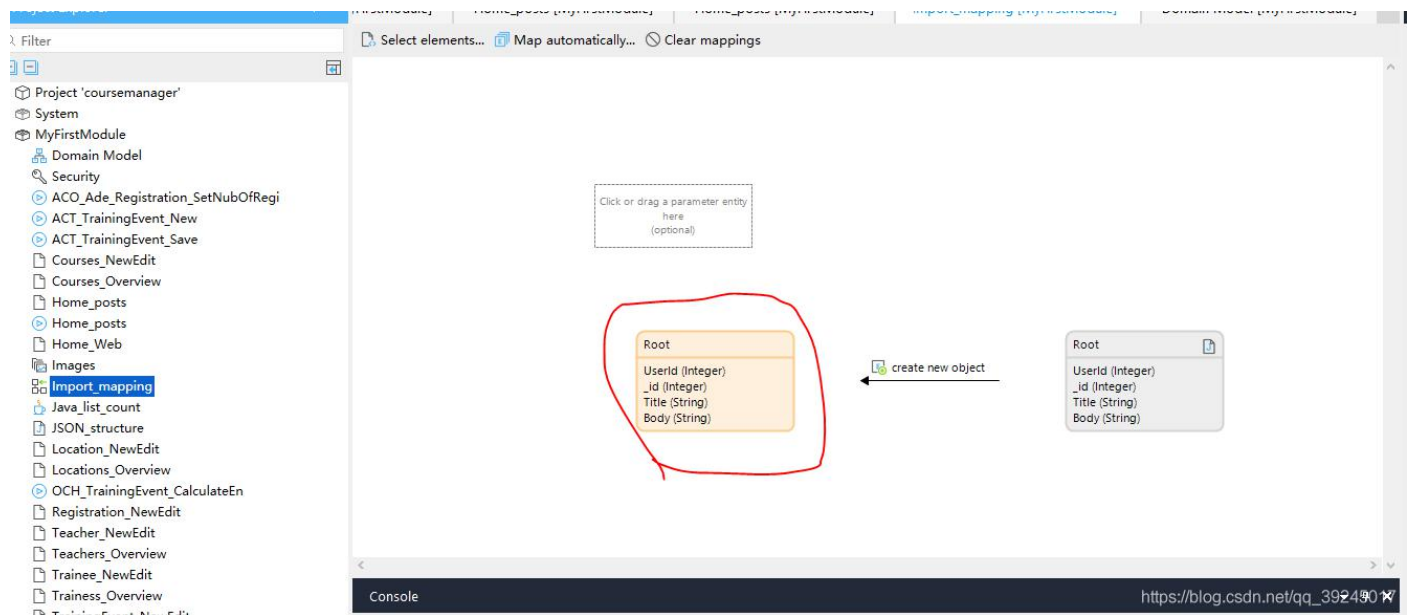
15. 创建好以后我们需要与我们的数据模型进行关联，然后点击下图的按钮进行自动创建并关联，点击 close 按钮。



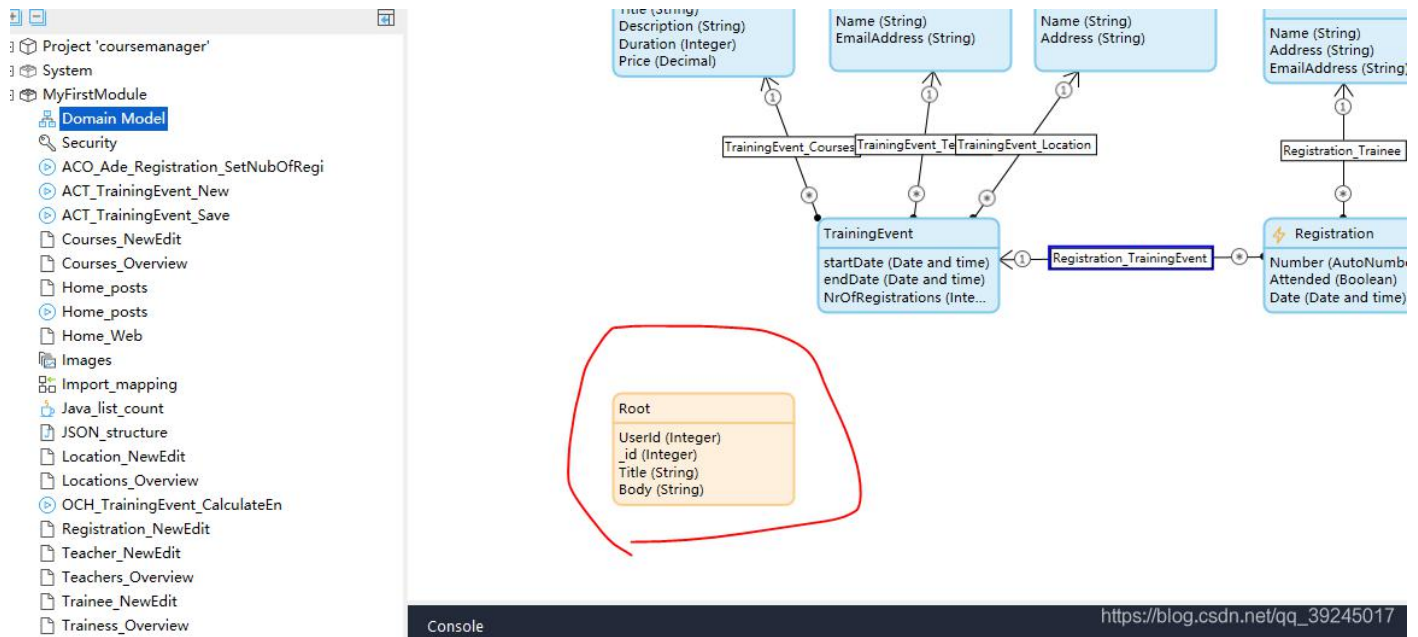
https://blog.csdn.net/qq_39245017



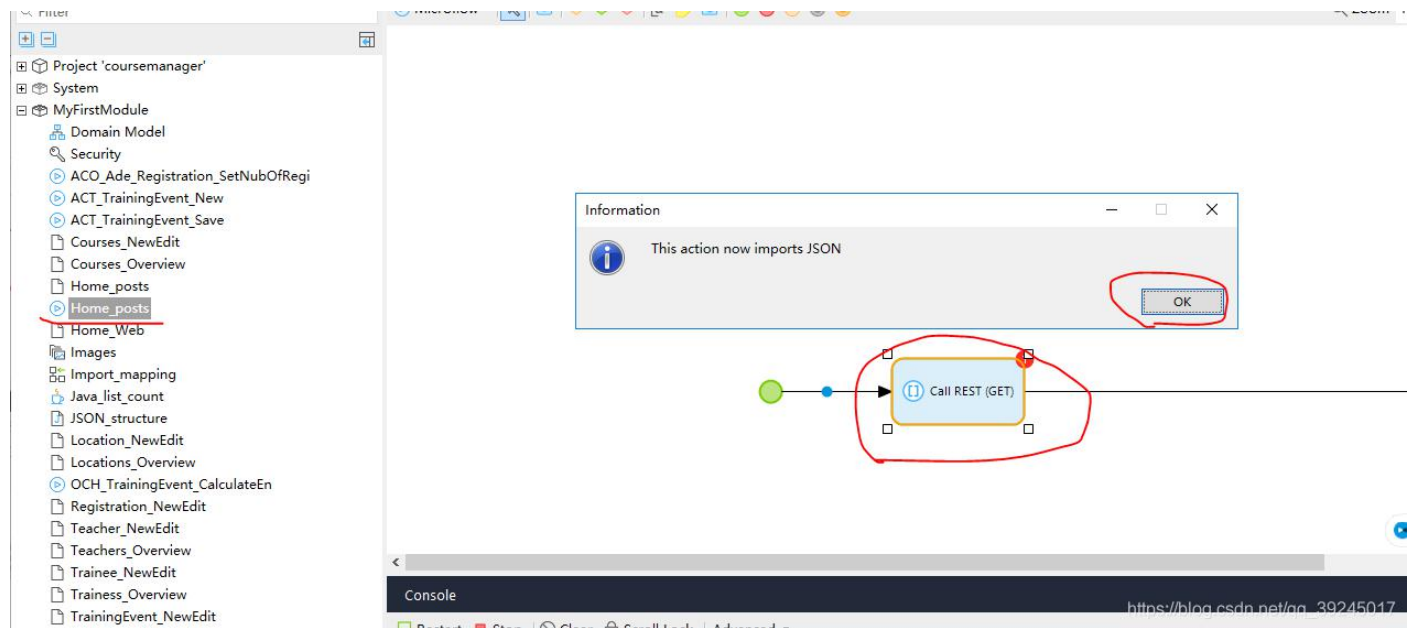
16. 系统会自动的为我们创建一个内存缓存数据模型。



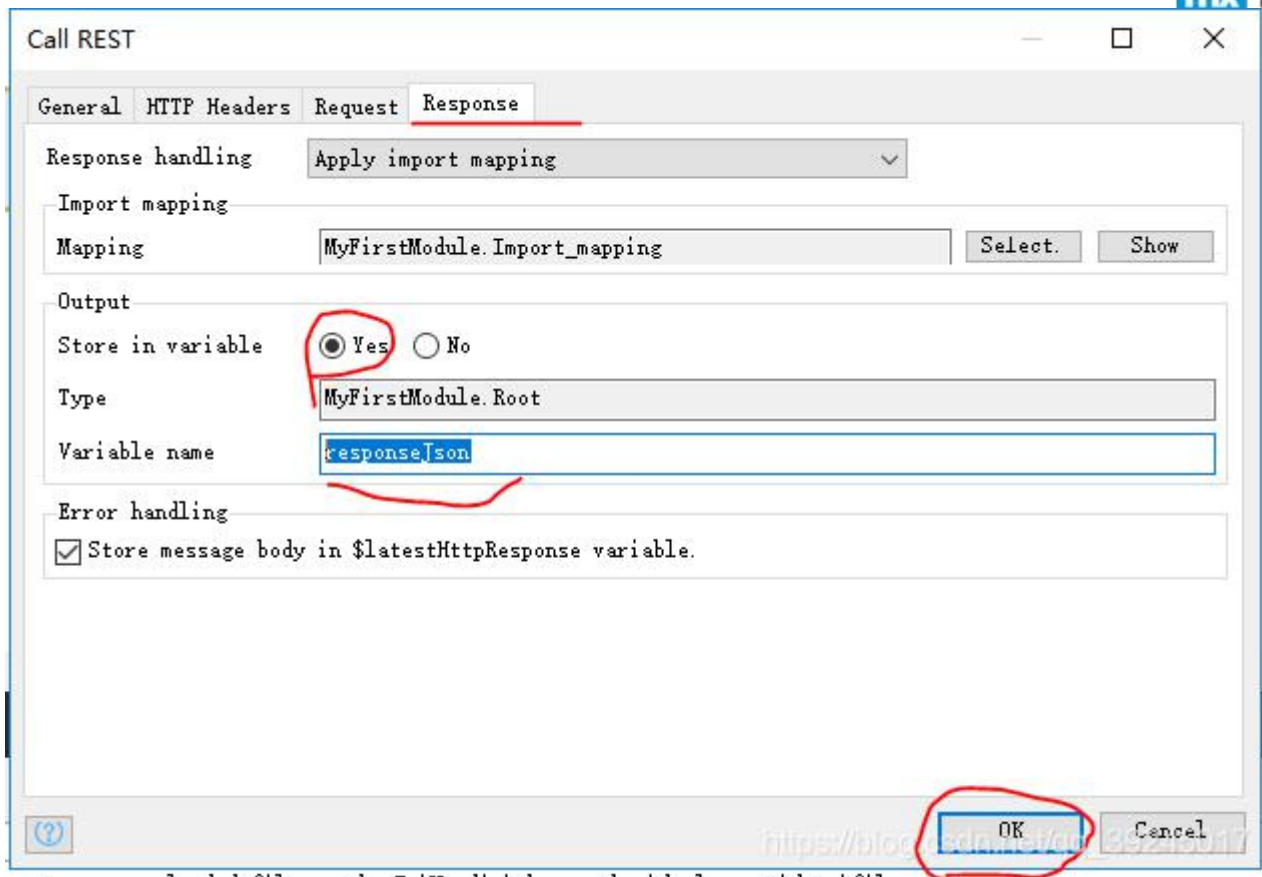
17. 我们同样的也可以在我们的数据模型工作台中看到创建的这个临时 Root 数据模型。



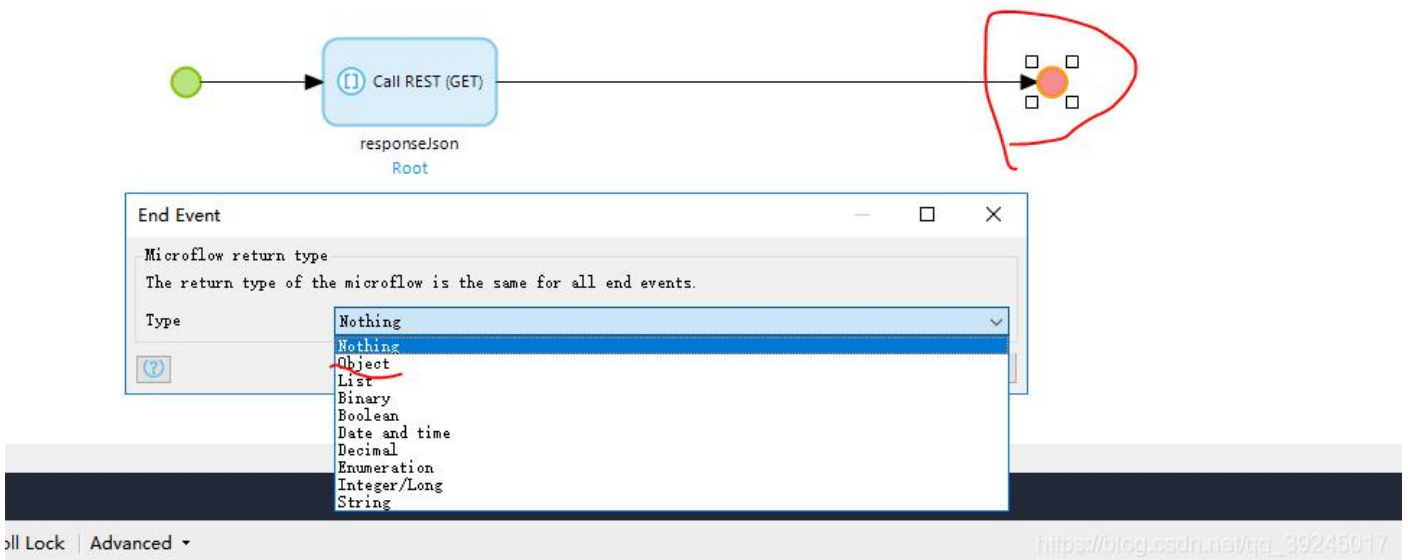
18. 接下来我们需要点击我们的微流，然后双击报错信息。点击 Ok 按钮，进行设置。



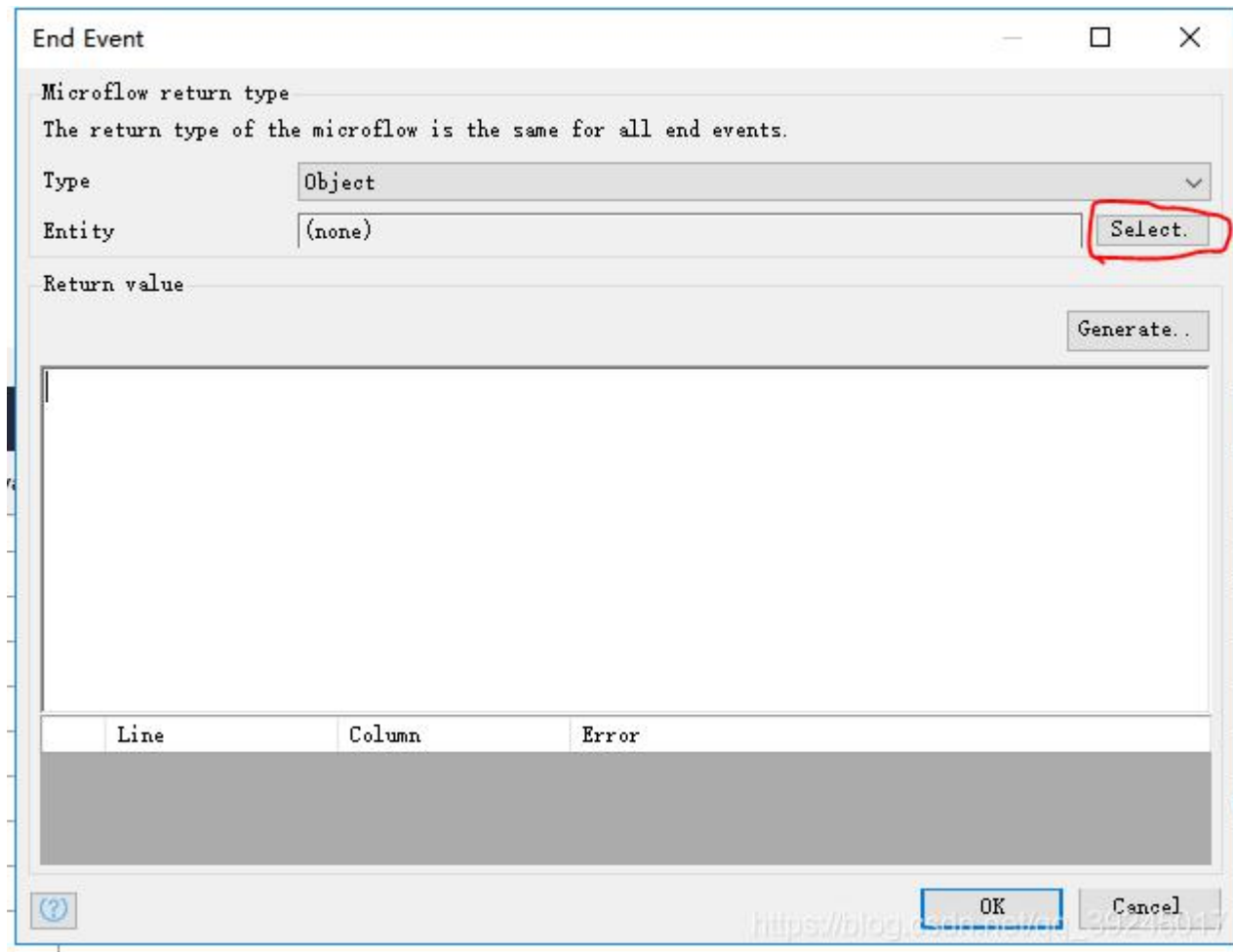
19. 这时候我们需要输出相应体，然后在 Store in variable 选择 Yes 按钮，输入返回的实体名称 responseJson，然后点击 Ok 按钮。



20. 我们双击结束按钮，在弹出的页面我们选择 Object。



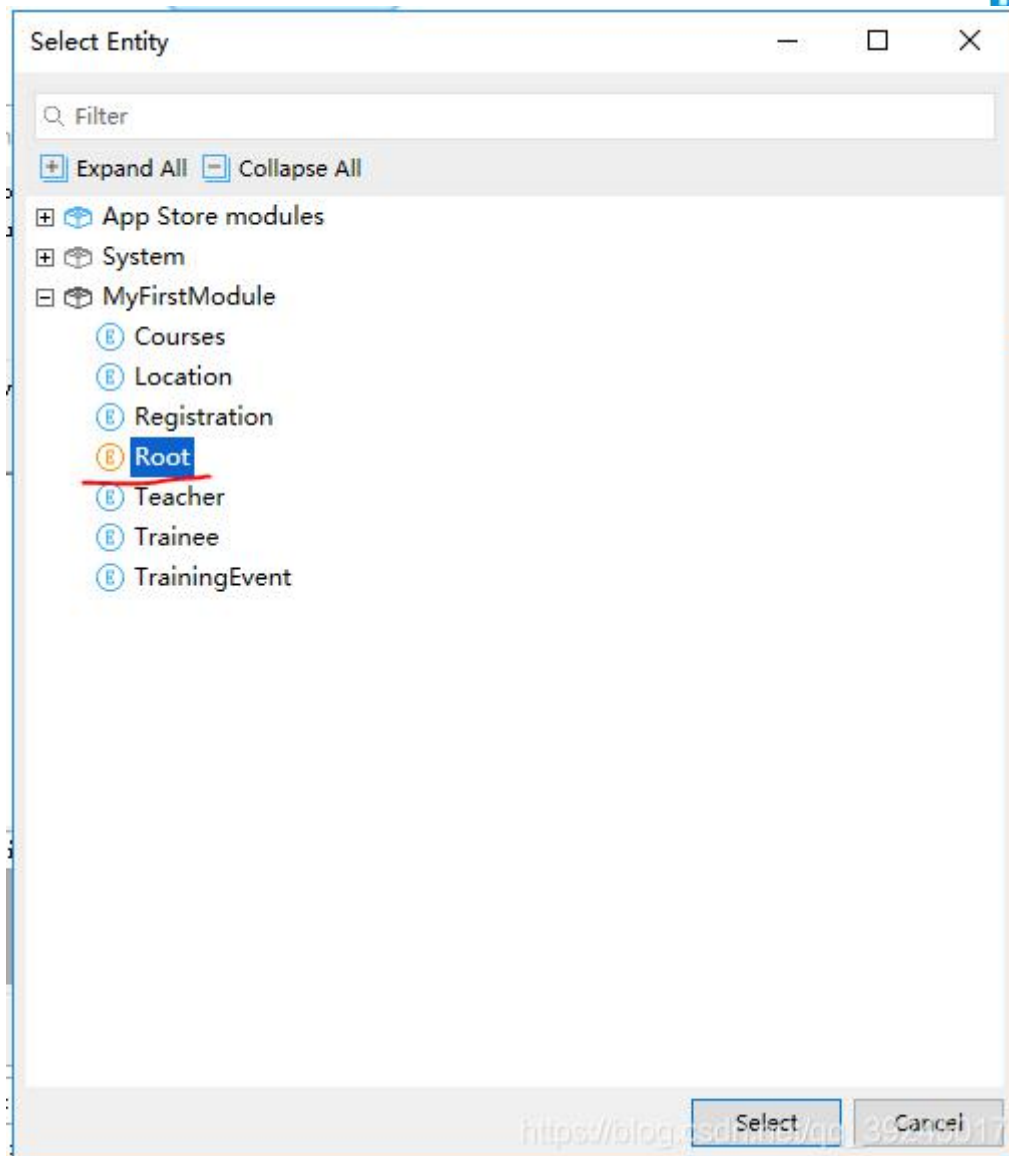
21. 然后选择返回的数据模型， 点击 Select 按钮。



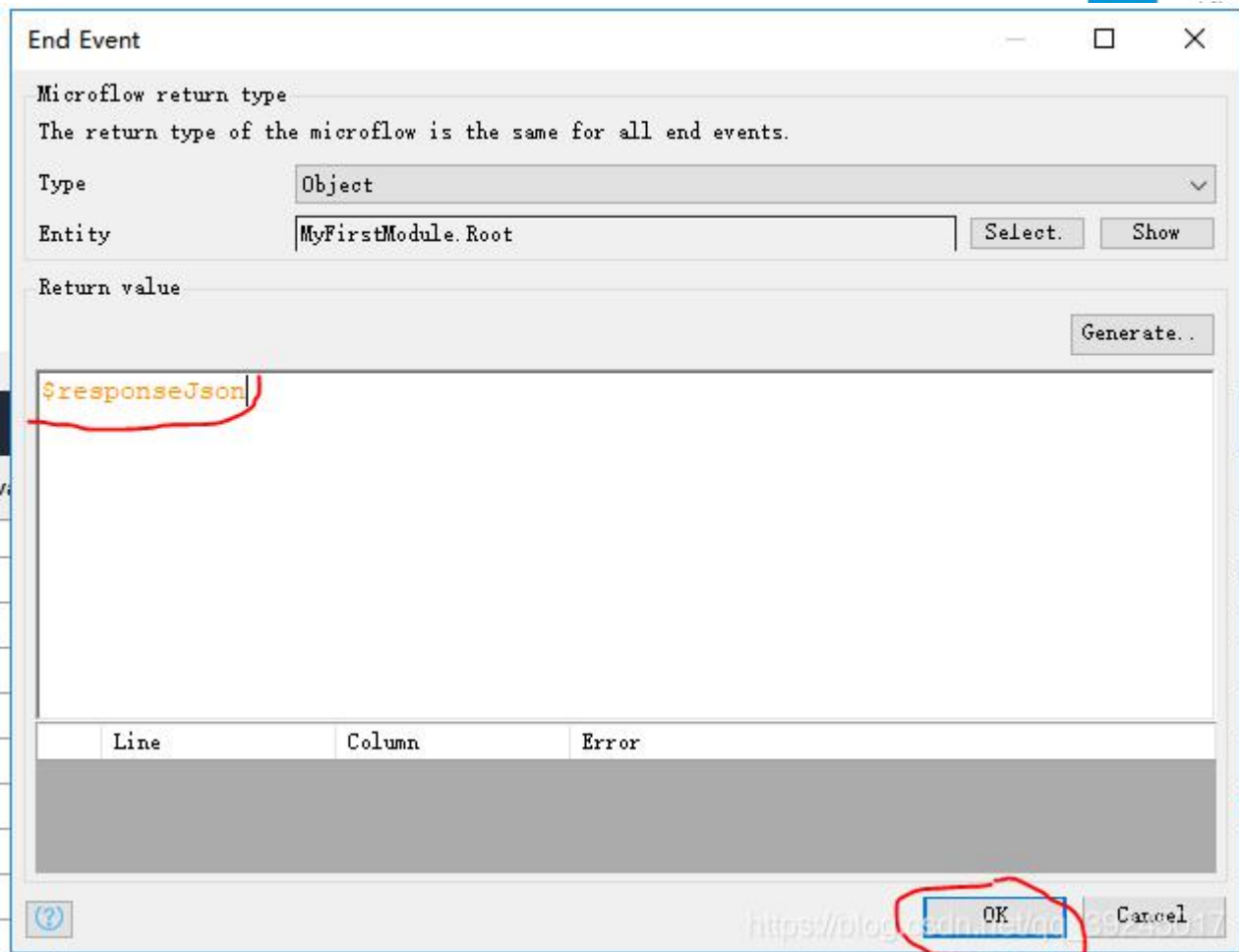
The image shows the 'End Event' dialog box in Mendix. It has a title bar with a close button. The main content area is divided into sections. The first section is 'Microflow return type' with the text 'The return type of the microflow is the same for all end events.' Below this is a 'Type' dropdown menu set to 'Object' and an 'Entity' dropdown menu set to '(none)'. A red circle highlights the 'Select.' button next to the 'Entity' dropdown. Below these is a 'Return value' section with a 'Generate..' button. At the bottom is a table with columns 'Line', 'Column', and 'Error'. The table is currently empty. At the very bottom are 'OK' and 'Cancel' buttons. A URL 'https://blog.csdn.net/qq_35223047' is visible in the bottom right corner.

Line	Column	Error
------	--------	-------

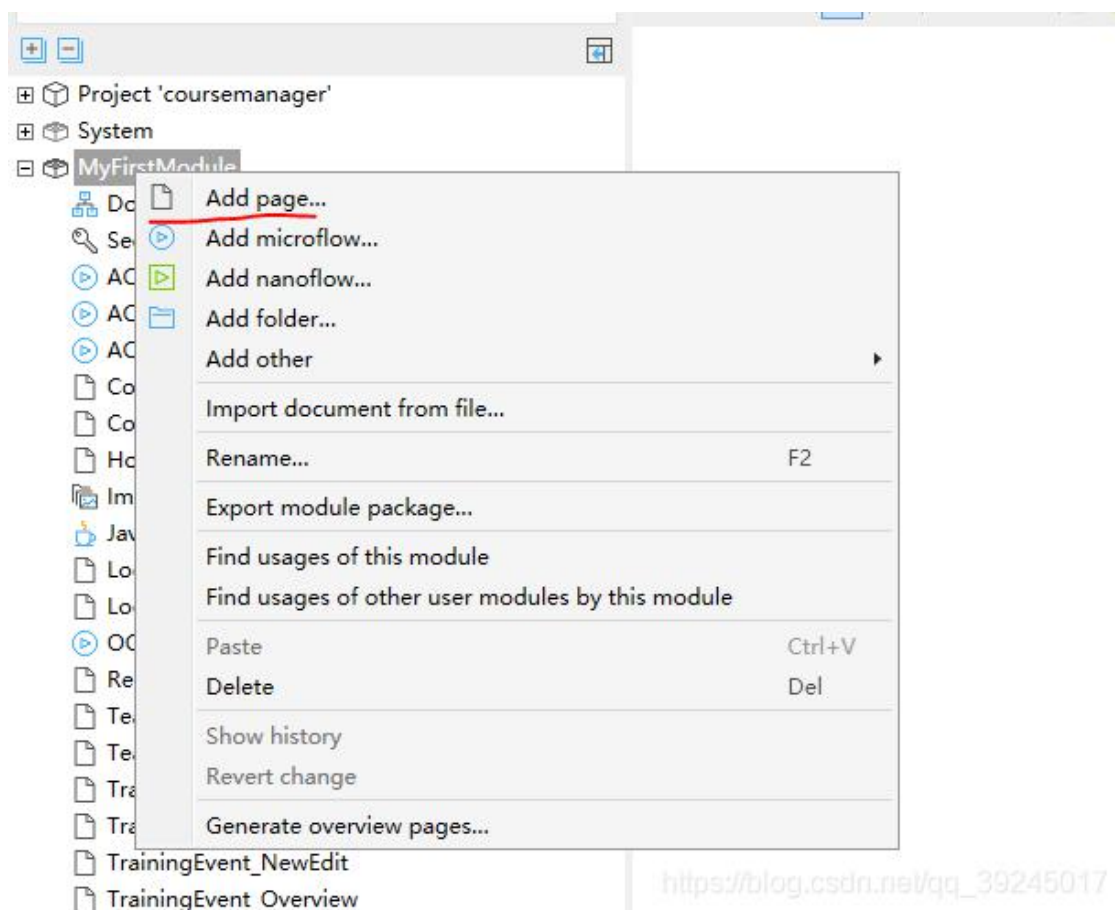
22. 然后选择新创建的 Root， 点击 Select 按钮。



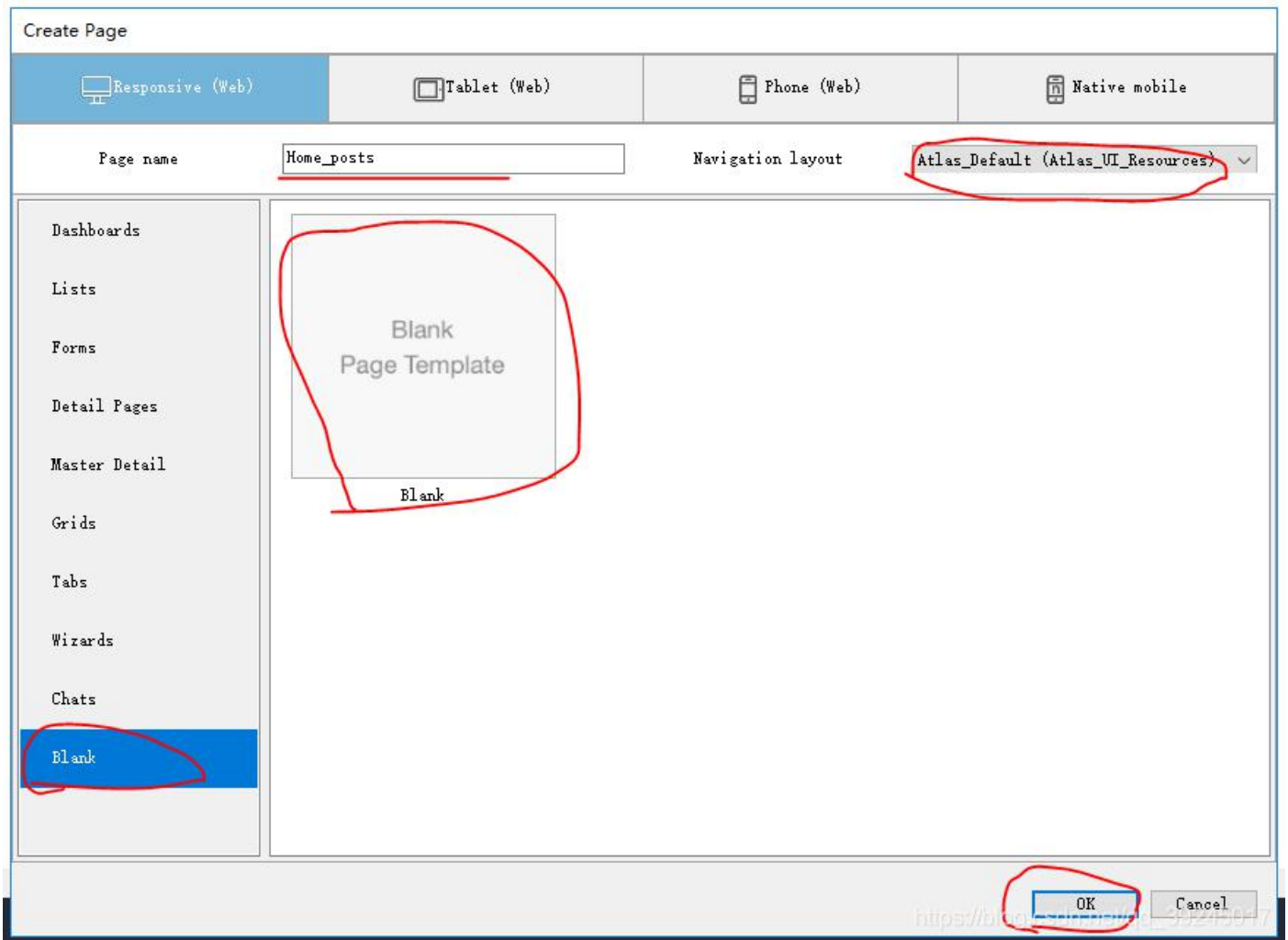
23. 然后给他赋值为我们刚创建的那个输出参数。点击 Ok 按钮。



24. 我们这个时候需要在页面展示，所以点击 Add page 新创建一个页面。



25. 我们创建一个空白的页面，命名为 Home_posts，然后点击 Ok 按钮。



Create Page

Responsive (Web) Tablet (Web) Phone (Web) Native mobile

Page name: Home_posts Navigation layout: Atlas_Default (Atlas_UI_Resources)

Blank Page Template

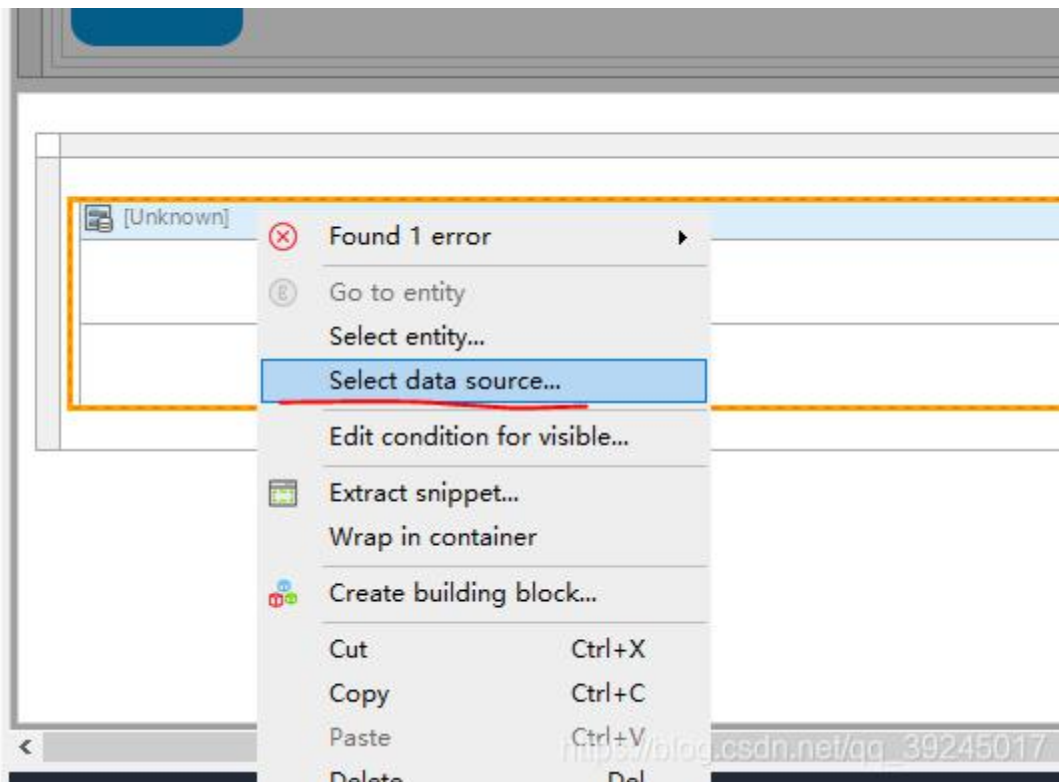
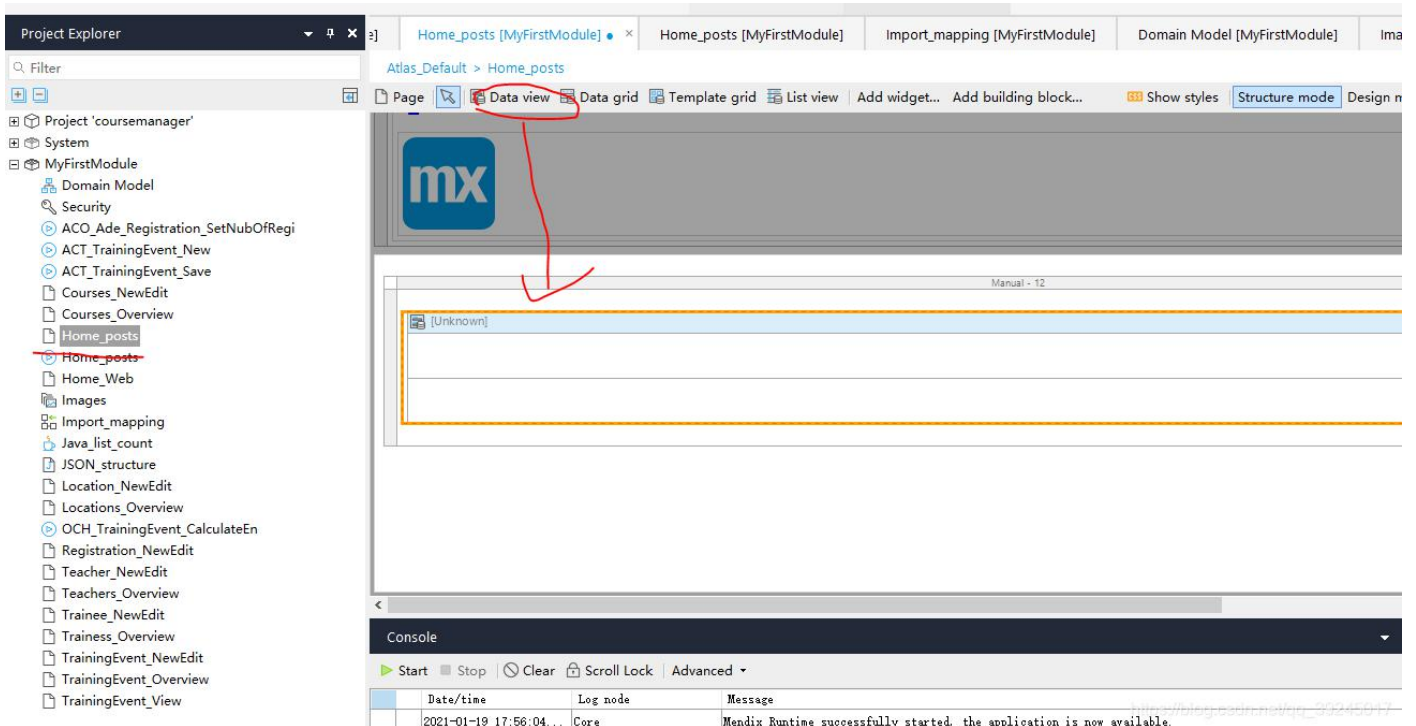
Blank

Blank

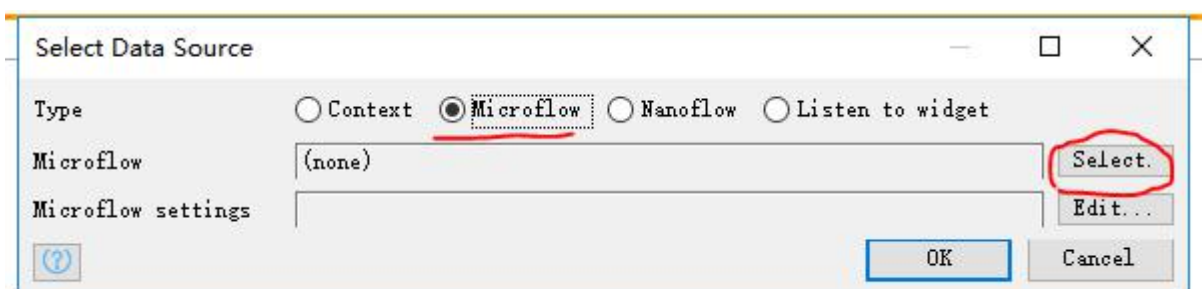
OK Cancel

<https://blogs.mendix.com/2017/02/24/5917>

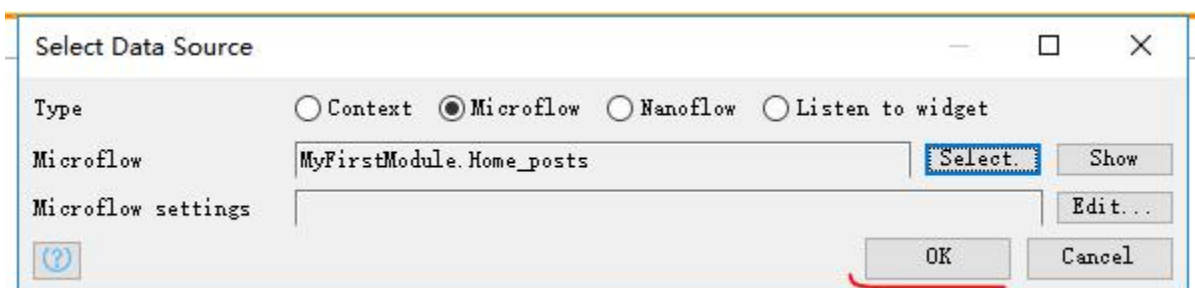
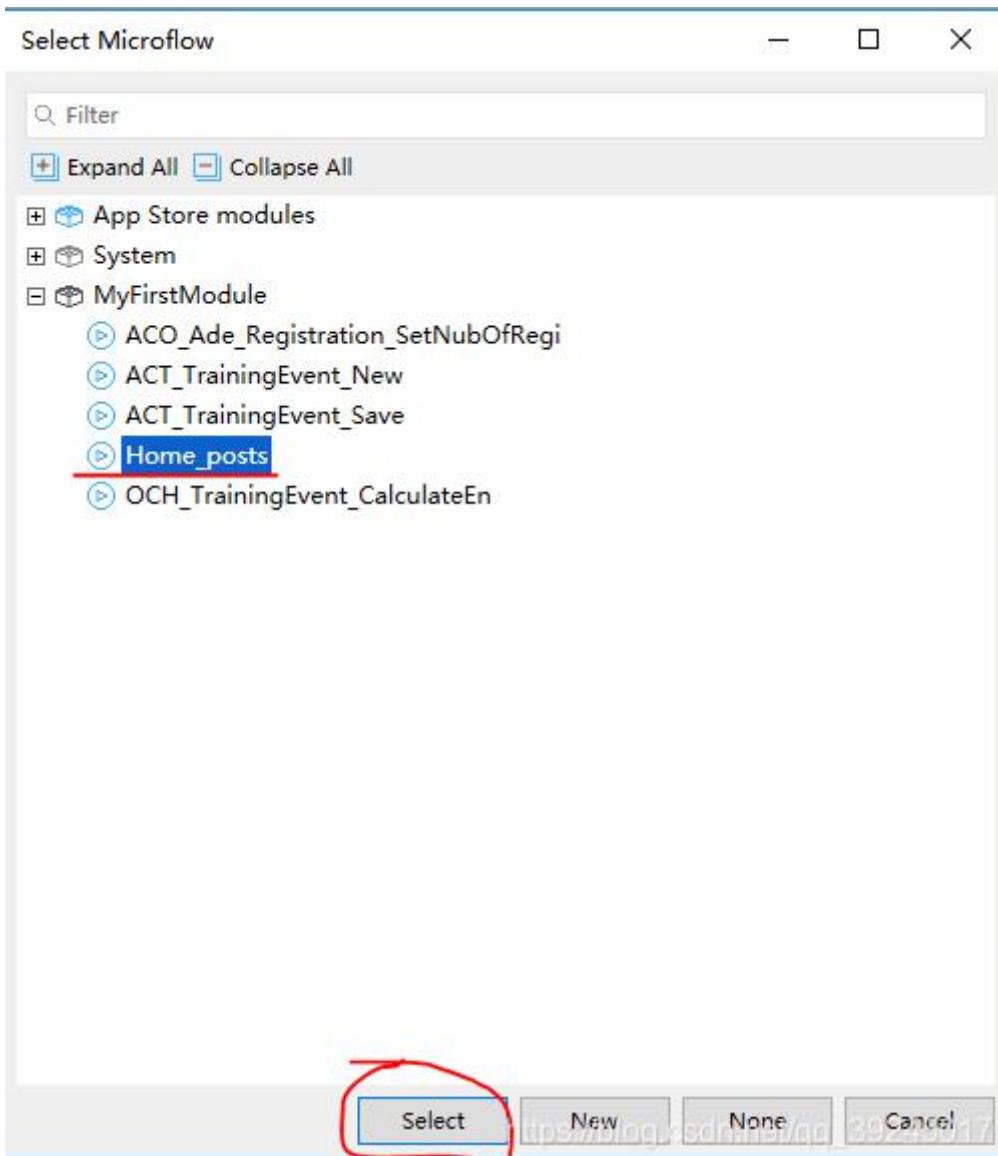
26. 拖入一个 Data view 组件。并右击选择数据源。



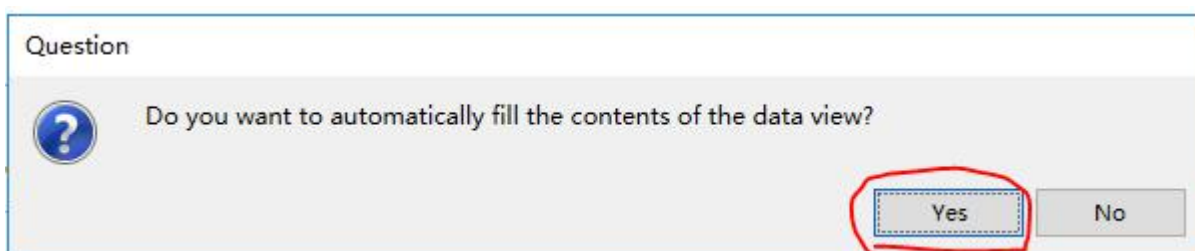
27. 这时候需要注意，我们要选择 Microflow 微流，然后点击 Select 按钮。



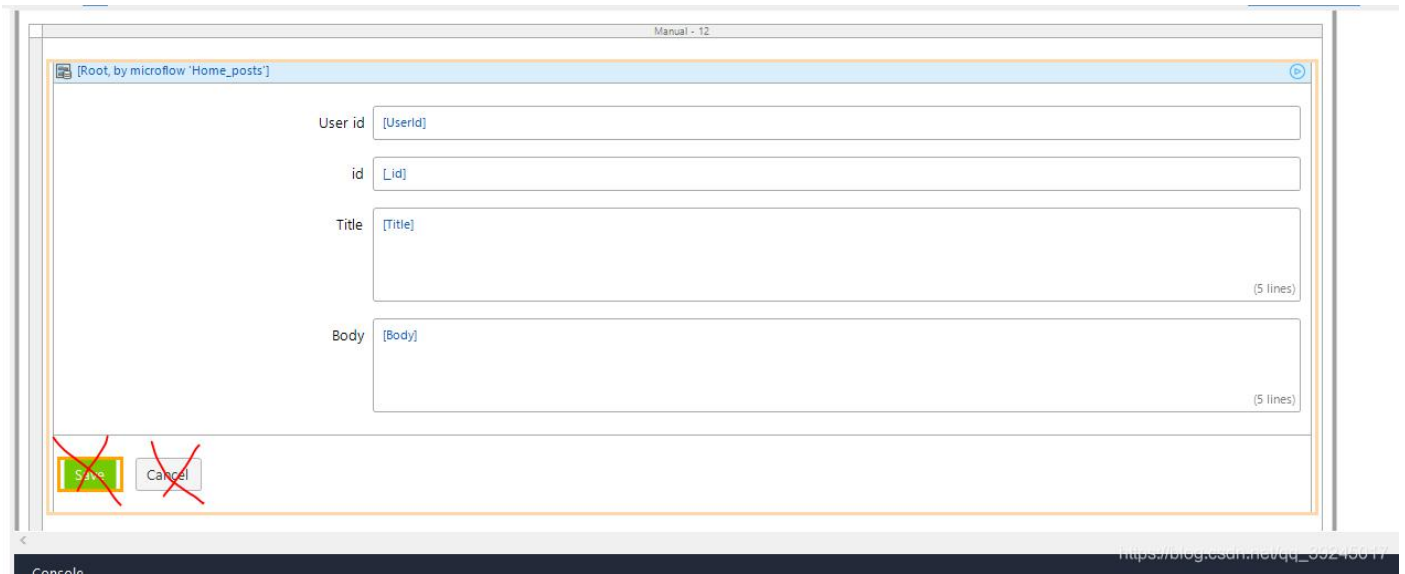
28. 然后选择我们的微流，点击 select->Ok 按钮。



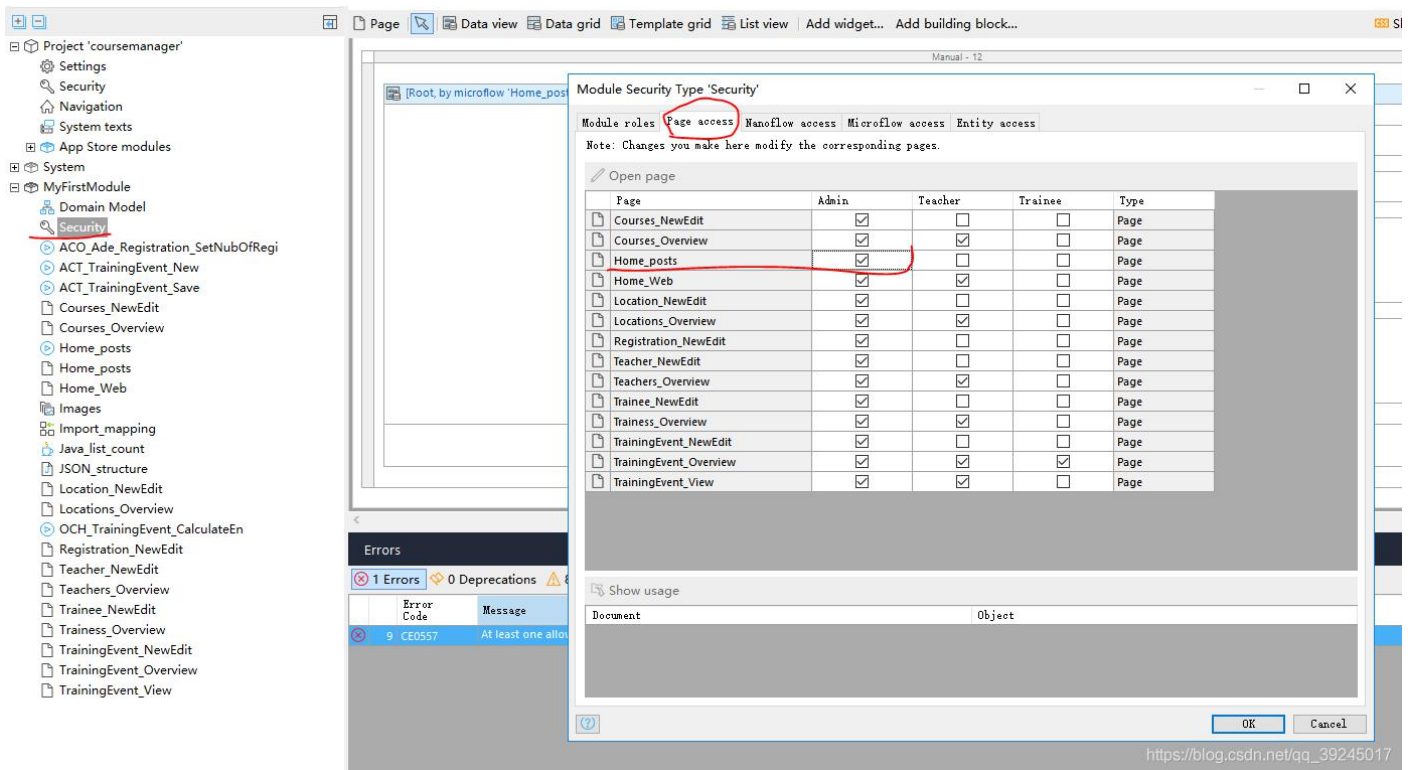
29. 我们选择 Yes，根据数据模型自动展示。



30. 删除不需要的按钮。



31. 我们现在需要模块的配置权限，在 Page access 配置 Admin 可见。



32. 在 Microflow 配置 admin 可用。

Module Security Type 'Security'

Note: Changes you make here modify the corresponding microflows.

Open microflow

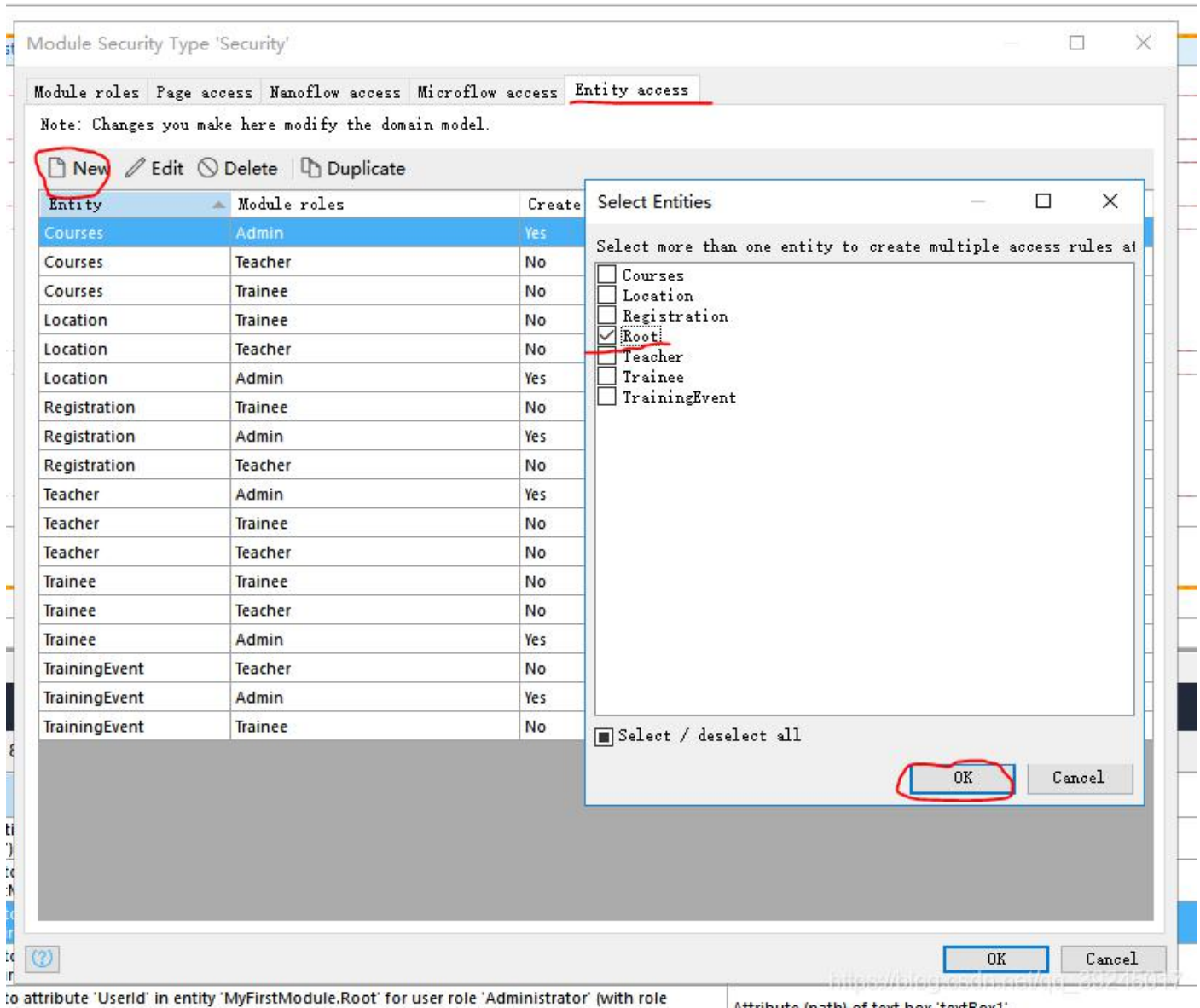
Microflow	Admin	Teacher	Trainee
ACT_TrainingEvent_New	☒	☐	☐
ACT_TrainingEvent_Save	☒	☐	☐
Home_posts	☒	☐	☐
OCH_TrainingEvent_CalculateEn	☒	☐	☐

Show usage

Document Object

OK Cancel

33. 我们在 Entity access 配置管理员可以创建删除，可读可写的权限。然后点击 Ok 按钮。



New Access Rule of Entity 'MyFirstModule.Root'

Documentation

Rule applies to the following module roles

- ☒ Admin
- ☐ Teacher
- ☐ Trainee

☐ Select / deselect all

Access rights XPath constraint

Create and delete rights

- ☒ Allow creating new obj
- ☒ Allow deleting existing objects

Member read and write rights

Default rights for new members ☒ None ☐ Read ☐ Read, Write

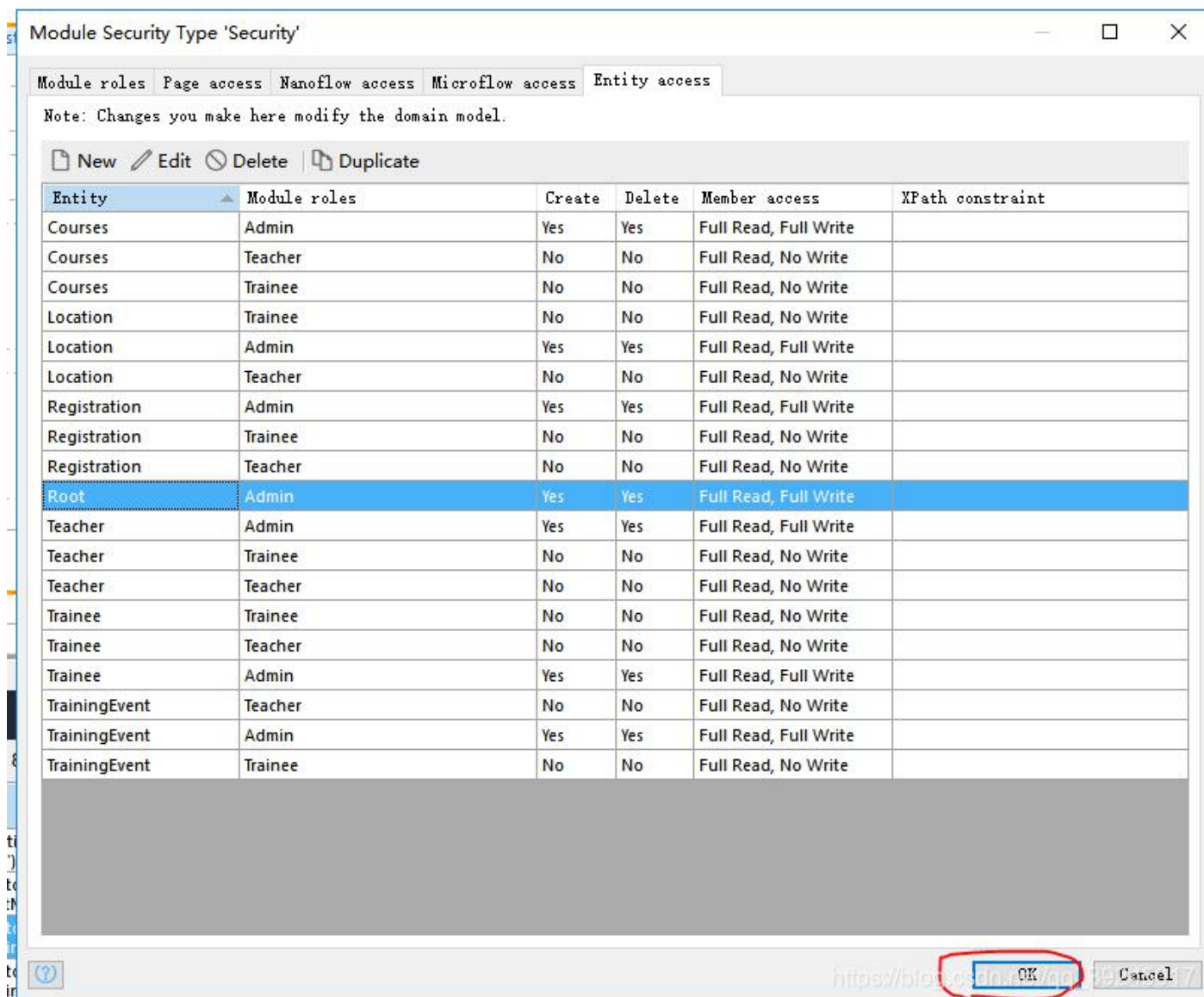
Member	Access rights
UserId (Integer)	Read, Write
_id (Integer)	Read, Write
Title (String (unlimited))	Read, Write
Body (String (unlimited))	Read, Write

Set all to

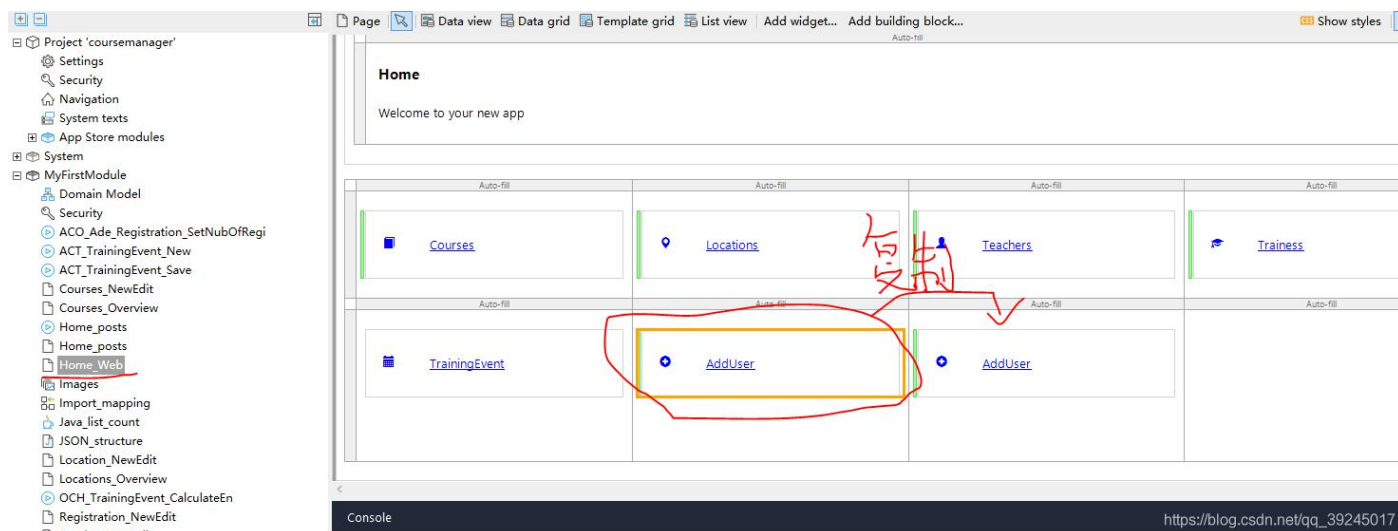
None Read **Read, Write**

OK Cancel

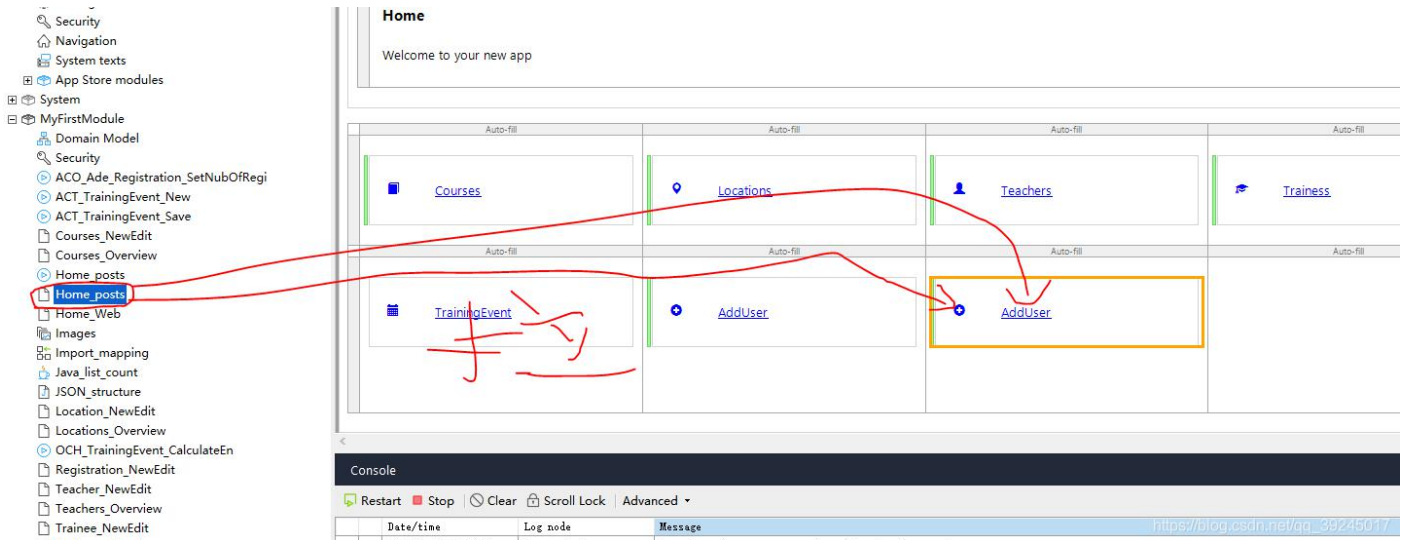
34. 配置好权限以后我们点击 Ok 按钮。



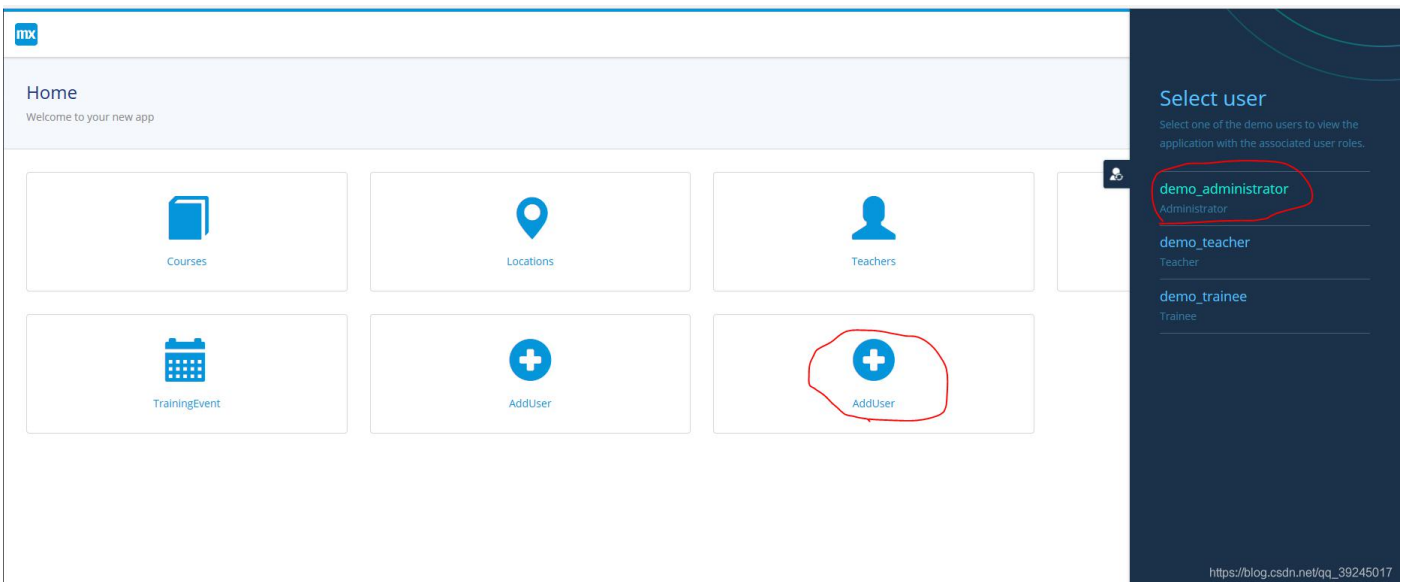
35. 我们在首页配置一个入口，直接复制下图圆圈中的组件。



36. 然后拖入我们的页面，直接拖就行。



37. 然后我们运行项目进行测试。发现已经实现接口的静态调用了。该功能到此就结束了。



User id	1
id	1
Title	sunt aut facere repellat provident occaecati excepturi optio reprehenderit
Body	quia et suscipit suscipit recusandae consequuntur expedita et cum reprehenderit molestiae ut ut quas totam nostrum rerum est autem sunt rem eveniet architecto

https://blog.csdn.net/qq_39245017