

# 06 - 模块 6：管理数据

## 模块 6：管理数据

**模块简介（模块图块）：**在此模块中，您将了解如何为您的应用程序添加一些交互。

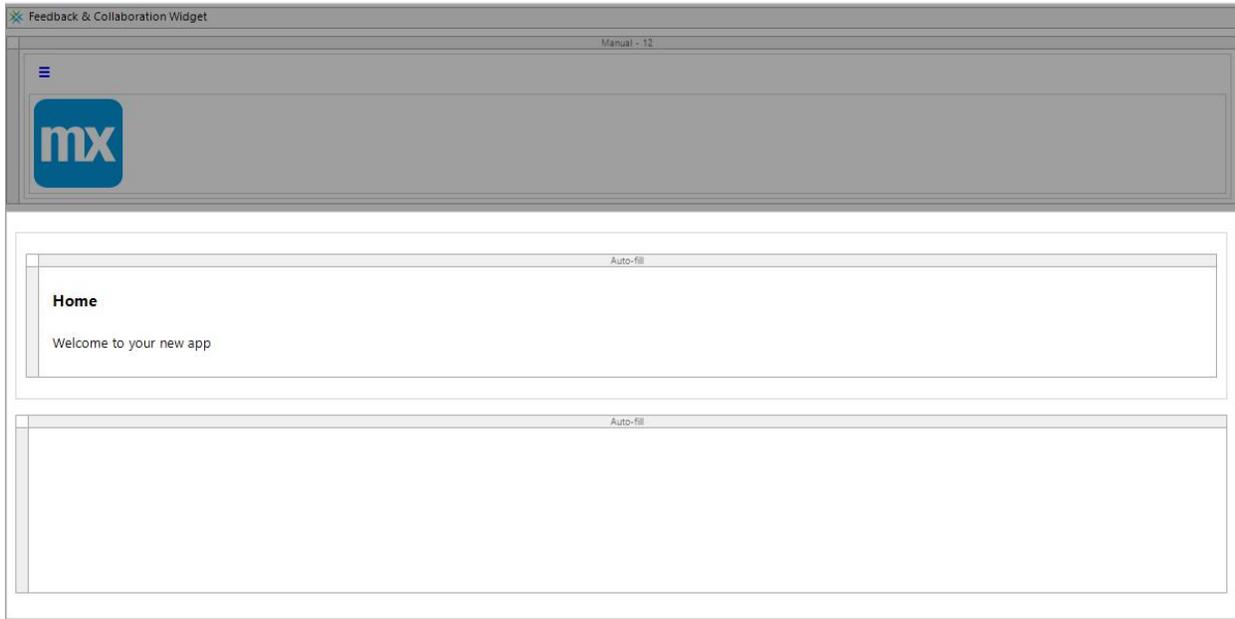
### 讲座 6.1：简介

后端配置完成后，现在是时候为我们的应用程序添加一些交互了。在 Mendix 中，我们使用页面编辑器实现此目标。您想要为 Summerhill 医院的人员创建页面，管理他们的用户和部门。他们应能够从 HR REST 服务导入新用户，设置和重置密码，并为外部人员创建用户帐户。数字化创新团队要求用单一页面来管理所有这些方面。他们对具体布局没有给出鲜明的偏好，因此您将使用默认的 Mendix 页面供他们启动管理。在他们看到第一个版本时，您就可以与他们讨论应用程序的 UX。Mendix 客户端遵循**模型、视图、控制器 (MVC)** 模式。**域模型**及其局部表示是**模型**的一部分。我们将在此模块中讨论另外两个部分。

### 讲座 6.2：Mendix 模板引擎

**MVC 的视图部分通过 Mendix 模板引擎**来管控。Mendix 提供基于 Web 的客户端，它的构建符合 JavaScript、CSS 和 HTML5 等现代 Web 标准。客户端实施采用 React 和 SCSS 等技术。Mendix 在 Studio Pro 中提供了页面构建器，可直观地提取这些技术为您所用。如此一来，即可轻松快速地创建页面。如果默认页面无法达到您的要求，您始终可以使用同一编辑器创建您的专属布局。随后可在其他应用程序中重用这些布局。

在下方，您可以看到显示在 Studio Pro 中的页面。此示例中的版块是布局网格。这些组件可用于将页面分成几列。每列可被设为三个值之一：自动填充、自动适合内容或手动。对于手动，最多可以使用 12 列。客户端使用基于 Twitter Bootstrap 的 CSS 来显示页面。它的优势在于确保您的页面响应及时。



在后台，Mendix 为您的页面生成 XML 文件。页面从布局开始，这里是 Atlas\_Default 布局。此布局接受一个参数。此参数包含我们在页面编辑器中看到的页面的 HTML 表示。它由 HTML 元素组成。这些元素可以充当普通 HTML 元素或组件的占位符。生成页面时，模板引擎将用适当的 HTML 替换这些占位符。

```
<?xml version="1.0" encoding="utf-8"?>
<m:page id="7b22948f-b2dd-43e1-9a8b-25011a7e1b33"
  xmlns="http://www.w3.org/1999/xhtml" title="Homepage" class="layout-atlas layout-atlas-responsive-default page-dashboard page-dashboard-actions"
  xmlns:m="http://schemas.mendix.com/foras/1.0">
  <m:layouts>
    <m:layout path="Atlas_UI_Resources/Atlas_Default.layout.xml"></m:layout>
  </m:layouts>
  <m:arguments>
    <m:argument parameterName="33328f4e-9727-4539-822f-f25179397b74">
      <div class="mx-name-container1 pageheader">
        <div data-mendix-id="0_0" class="mx-layoutgrid mx-layoutgrid-fluid container-fluid mx-name-layoutGrid2">
          <div class="row">
            <div class="col-md-12">
              <div data-mendix-id="0_1" data-mendix-type="mxui.widget.ReactWidgetWrapper" data-mendix-props="widgetTree:{{'widget':'com.mendix.widget.core.Text'}}">
                <div data-mendix-id="0_2" data-mendix-type="mxui.widget.ReactWidgetWrapper" data-mendix-props="widgetTree:{{'widget':'com.mendix.widget.core.Text'}}">
                </div>
              </div>
            </div>
          </div>
        </div>
        <div data-mendix-id="0_3" class="mx-layoutgrid mx-layoutgrid-fluid container-fluid mx-name-layoutGrid1">
          <div class="row">
            <div class="col-md-12">
              <div data-mendix-id="0_4" data-mendix-type="mxui.widget.ReactWidgetWrapper" data-mendix-props="widgetTree:{{'widget':'com.mendix.widget.core.ActionButton'}}">
              </div>
            </div>
          </div>
        </div>
      </m:argument>
    </m:arguments>
  </m:page>
```

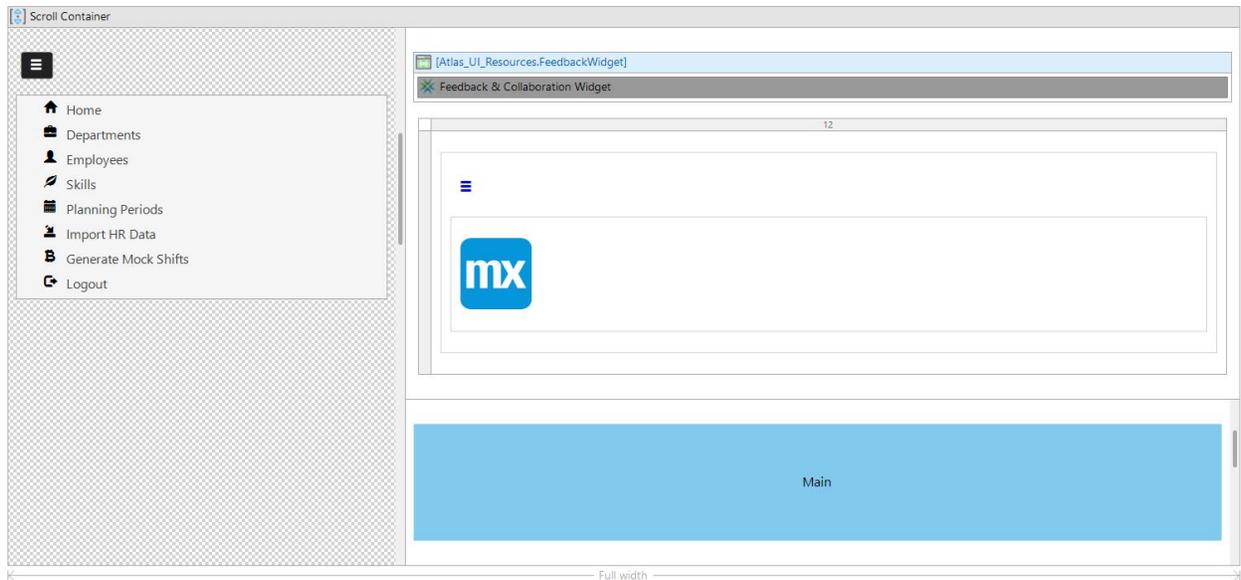
页面存储为 XML 信封，其中包含 HTML 和指向可能的布局文档的链接。文件将按层次结构进行评估，从页面级别开始。参数元素中的 HTML 内容将传递到布局元素中指定的文件。该布局将具有一个或多个将接收这些参数的占位符。客户端将解析这些文件并渲染可发送到浏览器的 DOM 树。

### 讲座 6.3: 布局

布局组件本身也是 XML 文件，它们包含 HTML 元素。客户端通过下载适当的 XML 文件来构建页面。此文件由模板引擎解析。任何其他布局文件都下载并合并到页面中。此过程将重复，直到下载所有布局，且最终页面在浏览器中可见。

XML 文件具有页面的根元素。此元素可以具有多个布局元素。这些布局元素对应于项目中的布局文档。每个布局都有占位符，可以使用参数元素中的元素填充这些占位符。这些元素包含您在 Studio Pro 创建的页面中。除此之外，模板元素还包含可用于生成带动态内容的页面的模板列表。

该页面引用了 **Mendix** 客户端将用来构建最终 DOM 树的布局。此布局文件可以在页面编辑器中查看，方法是单击 **编辑** → **查找...**，然后在 **Atlas 默认** 中键入。如果单击搜索结果，可看到下图。



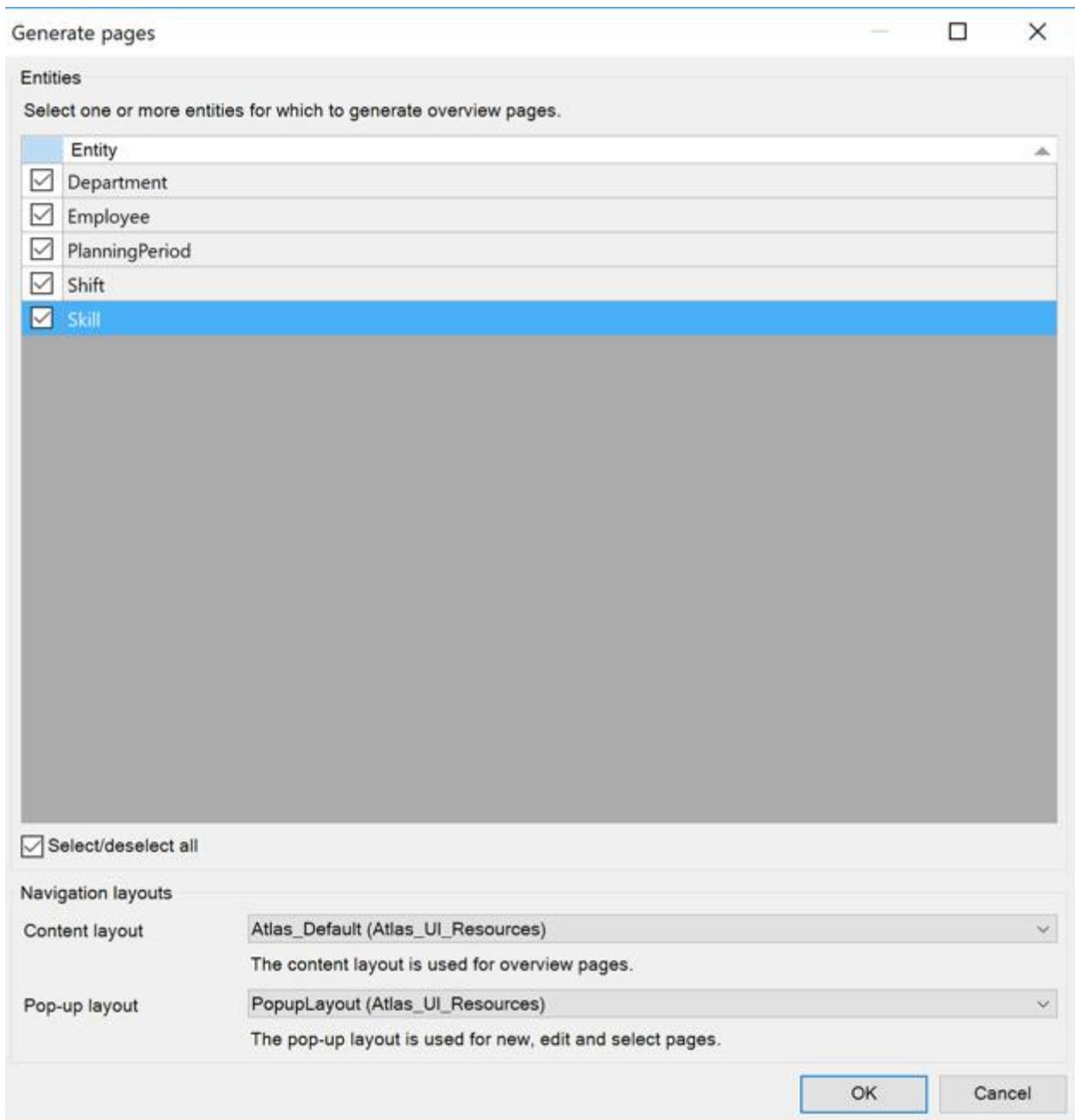
此布局也有 XML 表示，如下图所示。此布局和为您的页面创建的布局文件用于更新 DOM。

```
1 <mx:layout
2   xmlns="http://www.w3.org/1999/xhtml"
3   xmlns:mx="http://schemas.mendix.com/forms/1.0" id="2883277e-2832-4357-8e75-ec17d2b78c3">
4   <mx:arguments>
5     <mx:argument>
6       <div data-mendix-id="182_0" data-mendix-type="mxui.widget.HorizontalScrollContainer" data-mendix-props="{&quot;fixed&quot;:true,&quot;config&quot;:[{&quot;position&quot;:&quot;left&quot;,&quot;
7         <div class="mx-layoutcontainer-left mx-scrollcontainer-left region-sidebar" style="width:320px;">
8           <div class="mx-layoutcontainer-wrapper mx-scrollcontainer-wrapper">
9             <div data-mendix-id="182_1" data-mendix-type="mxui.widget.SidebarToggleButton" data-mendix-props="{&quot;friendlyId&quot;:&quot;Atlas_UI_Resources.Atlas_Default.sidebarToggle&quot;
10              <div data-mendix-id="182_2" data-mendix-type="mxui.widget.NavigationTree" data-mendix-props="{&quot;friendlyId&quot;:&quot;Atlas_UI_Resources.Atlas_Default.navigationTree&quot;,&
11              </div>
12            </div>
13          </div>
14        <div class="mx-layoutcontainer-center mx-scrollcontainer-center">
15          <div class="mx-layoutcontainer-wrapper mx-scrollcontainer-wrapper mx-layoutcontainer-nested mx-scrollcontainer-nested">
16            <div data-mendix-id="182_3" data-mendix-type="mxui.widget.VerticalScrollContainer" data-mendix-props="{&quot;fixed&quot;:true,&quot;config&quot;:[{&quot;position&quot;:&quot;top&quot;,&quot;
17              <div class="mx-layoutcontainer-top mx-scrollcontainer-top region-topbar">
18                <div class="mx-layoutcontainer-wrapper mx-scrollcontainer-wrapper">
19                  <div data-mendix-id="182_4_5" data-mendix-type="SprintFeedbackWidget.SprintFeedback" data-mendix-props="{&quot;friendlyId&quot;:&quot;Atlas_UI_Resources.FeedbackWidget
20                   <div data-mendix-id="182_6" class="mx-layoutgrid mx-layoutgrid-fluid container-fluid mx-name-layoutGrid">
21                     <div class="row">
22                       <div class="col-md-12">
23                         <div class="mx-name-container1 topbar-content">
24                           <div data-mendix-id="182_7" data-mendix-type="mxui.widget.SidebarToggleButton" data-mendix-props="{&quot;friendlyId&quot;:&quot;Atlas_UI_Resources.Atlas
25                            <div class="mx-name-container3 navbar-brand">
26                              <div data-mendix-id="182_8" data-mendix-type="mxui.widget.StaticImage" data-mendix-props="{&quot;friendlyId&quot;:&quot;Atlas_UI_Resources.Atlas_def
27                              </div>
28                            </div>
29                          </div>
30                        </div>
31                      </div>
32                    </div>
33                  <div class="mx-layoutcontainer-middle mx-scrollcontainer-middle region-content">
34                    <div class="mx-layoutcontainer-wrapper mx-scrollcontainer-wrapper">
35                      <div data-mx-placeholder="86dea1f-2e59-4d2a-a2b5-47b6330ce846" class="mx-placeholder"></div>
36                    </div>
37                  </div>
38                </div>
39              </div>
40            </div>
41          </div>
42        </mx:argument>
43      </mx:arguments>
44    </mx:layout>
```

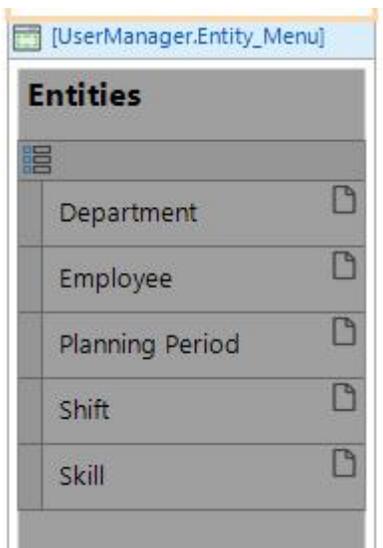
### 讲座 6.3.1: 生成页面

为了让开发人员的工作变得更轻松，Mendix 提供一种为您的实体生成页面的简捷方法。为了向 Summerhill 医院的用户提供基本用户界面，我们将使用此功能。生成页面后，我们可以进行更改，确保布局充分契合我们的期望。

1. 打开**域模型**页面，右键单击其中一个实体并选择**生成概述页面...**
2. 在**生成页面**窗口，选择所有实体。
3. 确保将**内容布局**设置为 **Atlas\_default**。
4. 确保将**弹出布局**设置为 **PopupLayout**。
5. 单击**确定**，开始生成页面。



您将看到一个新文件夹被添加到名为 **OverviewPages** 的项目中。此文件夹包含 11 个元素：两个**概述**页面，两个 **NewEdit** 页面和一个 **Entity\_Menu** 片段。该片段表示可重用的用户界面部分。如果打开任何**概述**页面，则会看到名为**片段调用**的小组件。这允许您将**片段**拖到页面中，因此只需创建一次这部分用户界面。



现在很简单！只需单击几下，即可生成用于编辑数据的页面。让我们看看接下来会发生什么。

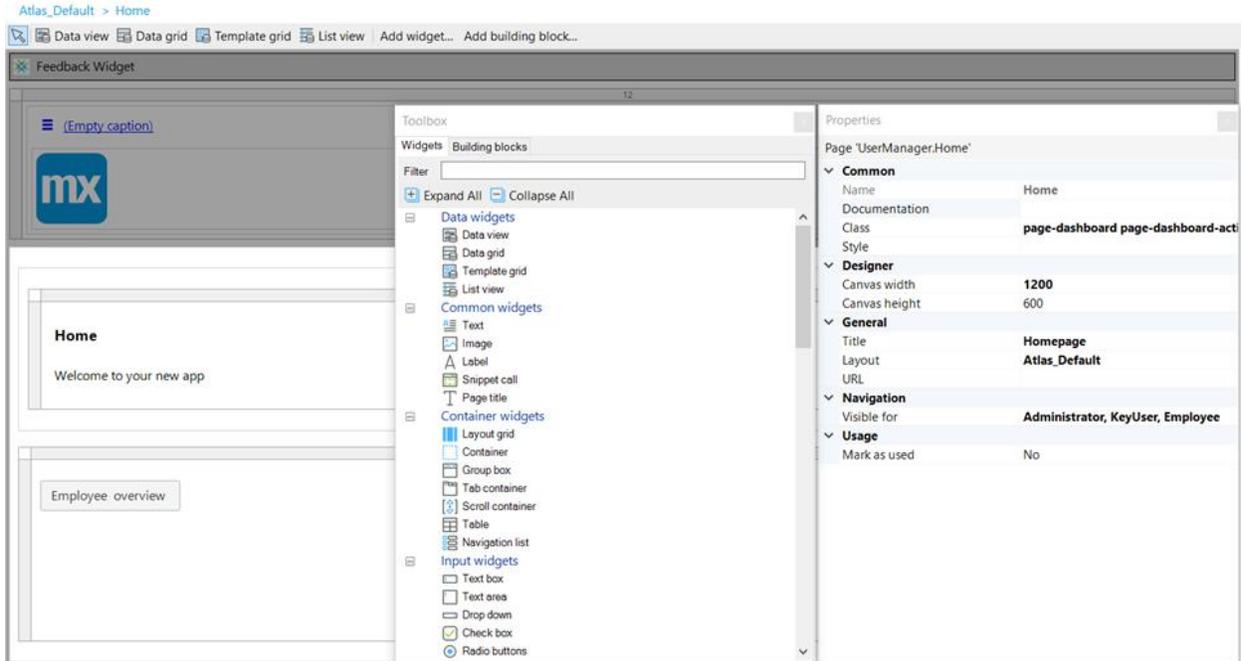
#### 讲座 6.4: 页面的命名规则

您可能已经注意到页面名称中的模式。这是因为我们也有针对这些页面的命名规则。下表为您提供这些命名规则的概览。

Page Purpose	Suffix
List objects of a single entity type	_Overview
Create an object	_New
Update an object	_Edit
Create <i>or</i> Update an object	_NewEdit
View an object (read-only)	_View
Select a single object	_Select
Select multiple objects	_MultiSelect
Tooltip	_Tooltip

#### 讲座 6.5: 页面编辑器

您已经知晓页面如何在后台构建，下面来看看如何使用页面编辑器，充分利用模板引擎的强大功能。此组件允许您使用小组件构建页面。默认情况下，页面编辑器包含许多小组件，可从市场添加小组件。如果找不到需要的小组件，您始终有机会编写自己的小组件并添加到项目中。



## 布局网格

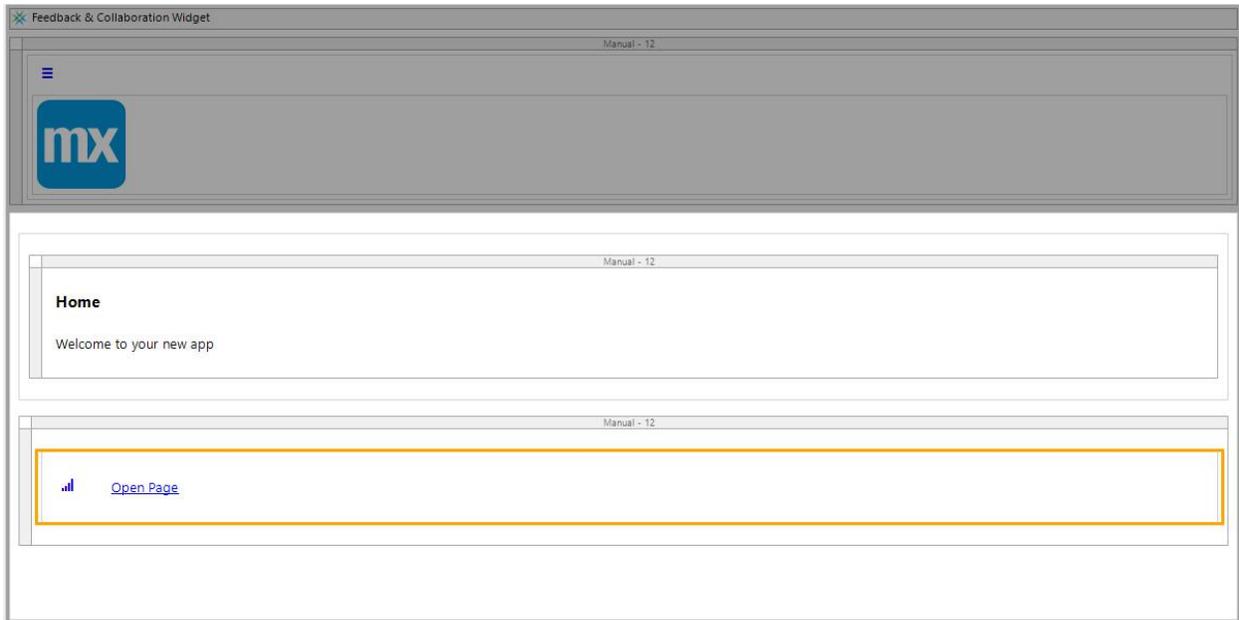
在下方，您可以看到显示在 Studio Pro 中的页面。此示例中的版块是**布局网格**。布局网格是一个小组件，它为页面提供结构。这些组件可用于将页面分成不同大小的列。客户端使用基于 Twitter Bootstrap 的 CSS 来显示页面。它的优势在于确保您的页面响应及时。布局网格包含一行或多行，每行都包含一列或多列。行数和列数没有限制。行将垂直显示，每行所占的垂直空间与该行中最大的一列一样多。列水平显示，每列可以根据其内容的需要占用尽可能多的水平空间（**自动适应内容**），或者它将得到一部分剩余的可用空间（**自动填充**）。

除此之外，还有第三种模式，您可以在其中手动指定列获得多少空间。这是**手动设置**。使用此设置，最多可以创建 12 列。每列的权重必须介于 1 和 12 之间，指示该列将获得多少水平空间。例如，权重 6 等于水平空间的 50%。每行的总重量不能超过 12，而您的每行总重量必须小于 12。这会将您的内容拉到右侧。

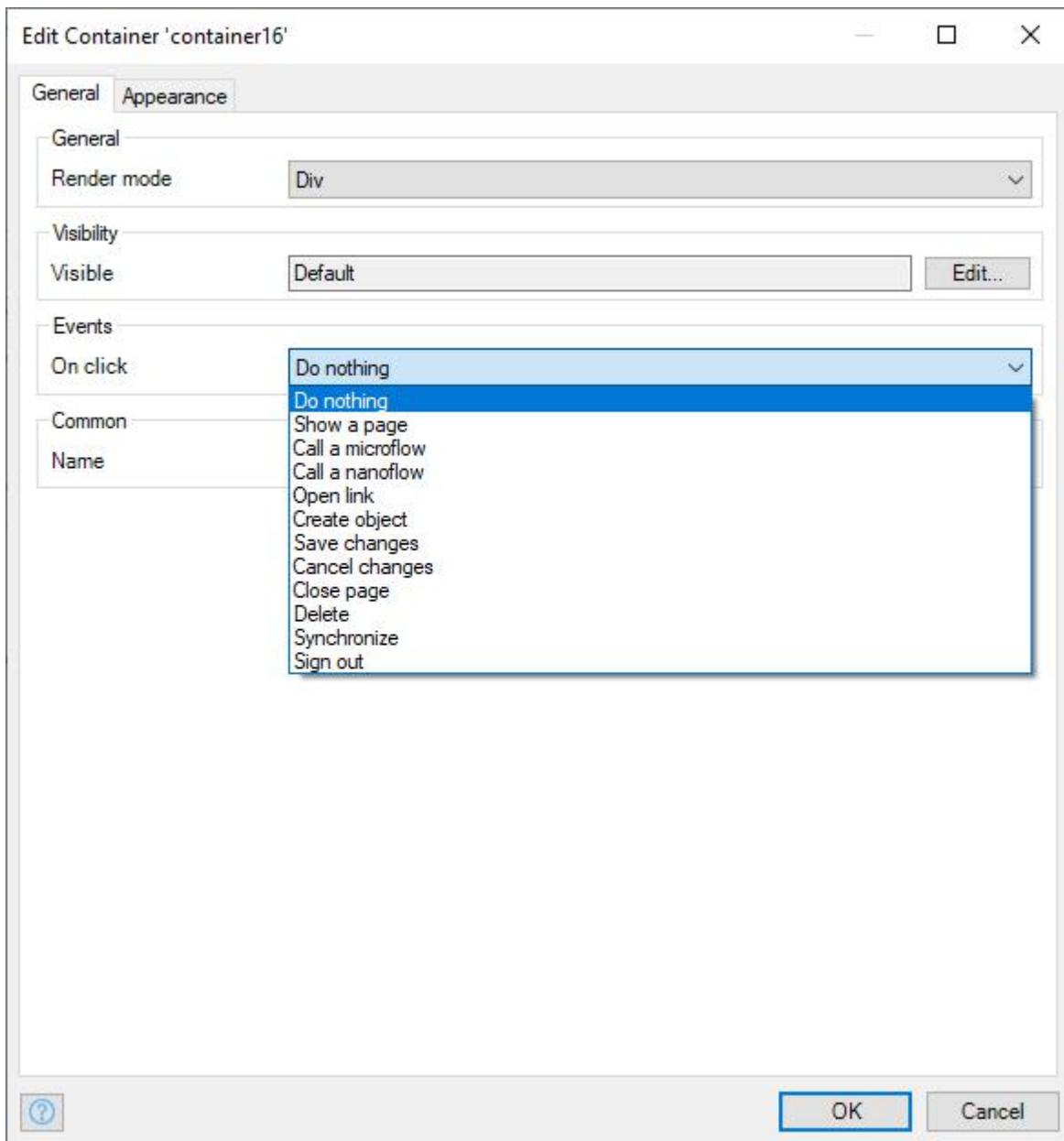
### 注意：

- 布局确定页面类型（特定于设备、响应式或弹出式）以及菜单位置。
- 布局网格是一个小组件，用来构建页面结构或布局。页面内容是否被分为多列？若是如此，这些列在各种设备类型上如何显示？

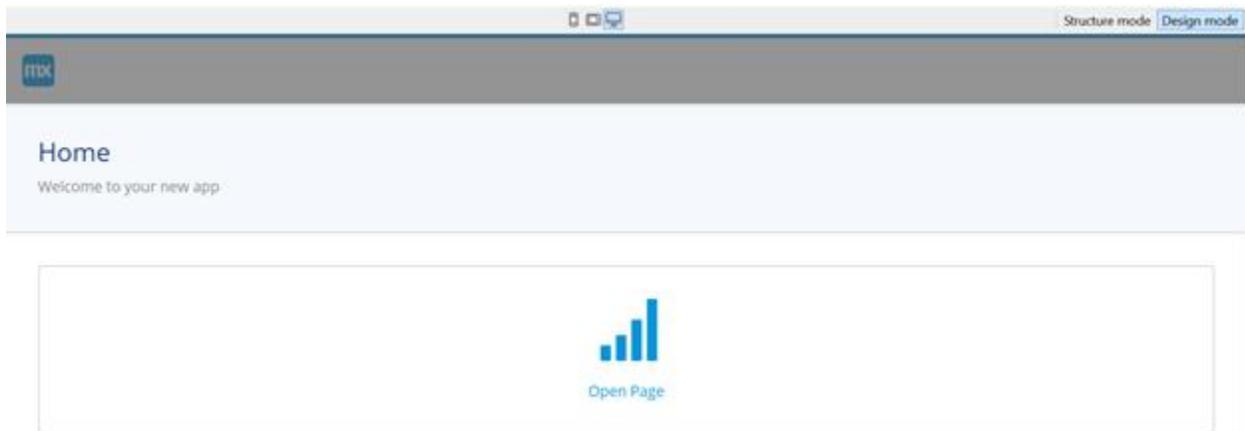
现在一秒钟试验页面编辑器。在主页 Home\_Web 上，从工具箱中拖出一个**卡操作**构建块，放在您的页面主体的某个位置。



可以看到，此构建块由三个小组件组成：标题、图标和周围容器。如果检查卡操作中任意元素的属性，您会发现有许多选项可使按钮具有独特性。其中一个较为重要的功能是**单击时**设置，可供您确定单击小组件将执行哪些操作。有许多默认行为，还有以微流或纳米流的形式执行定制逻辑的选项。

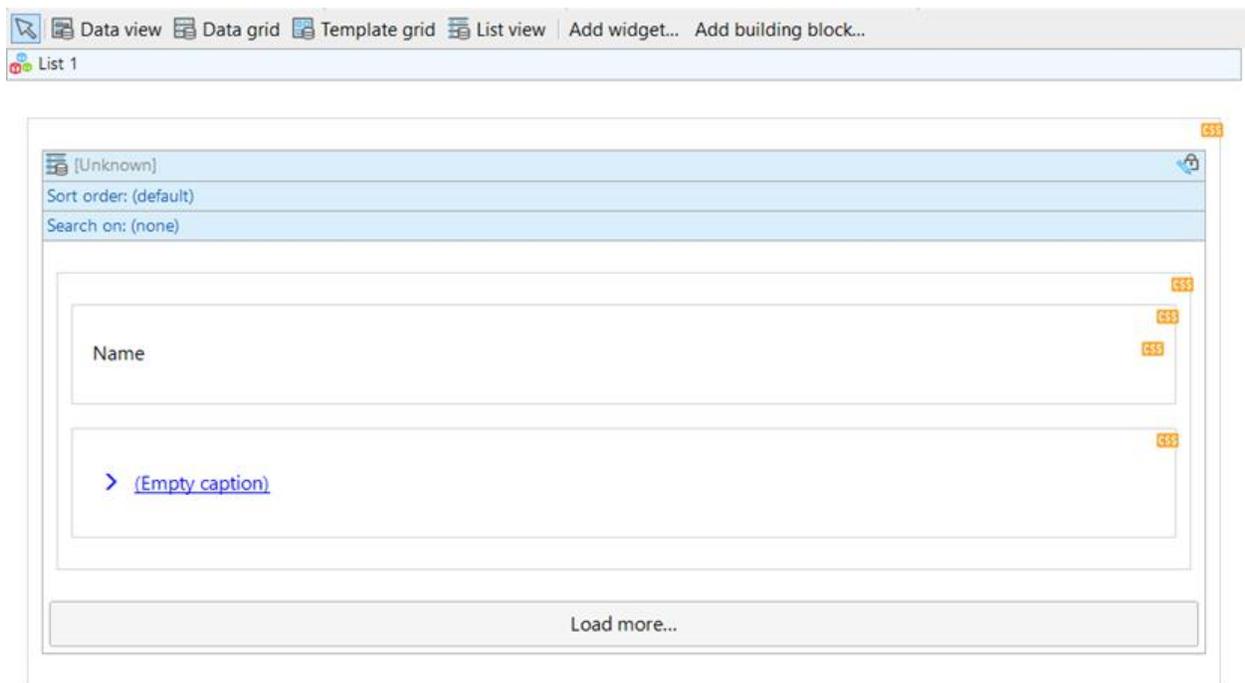


要在不部署应用程序的情况下在您的主页上查看外观，单击编辑器右上角的**设计模式**。这将显示 WYSIWYG 模式下的页面！卡操作将显示为外观迷人的超大按钮。



**结构模式**是 Studio Pro 中的默认编辑模式，清楚地显示页面的容器结构。在设计模式下，您可能会注意到，编辑器顶部有三个**特定于设备布局**选项。这些选项允许您查看应用程序在不同屏幕大小的设备上的显示情况，如有需要，还可以更改您的页面，使它们在每个设备上的显示方式不同。

完成试验后，删除**卡操作**并返回到“结构”模式，这样我们就可以了解在页面中显示数据的相关内容。



上图是更复杂的构建块的另一个示例，这次使用我们尚未讨论的小组件类型：**数据上下文**小组件。这种类型的小组件允许您对动态数据所需的布局进行构造，从而大幅缩减用户界面的开发时间。在下一个讲座中，我们将深入探讨有关小组件的详细信息。

## 讲座 6.6：数据上下文小组件

Mendix 页面编辑器的关键组件之一是**数据小组件**。这些小组件安排运行时和前端之间的连接，在客户端所基于的**模型、视图、控制器**模式中提供**控制器**部件。在 Mendix 中，我们将页面最初有权访问的数据称为上下文。上下文涉及由 Mendix 传递给页面的单个对象。

Mendix 在此类别中提供 4 个小组件。

- **数据视图**

- 此小组件用于显示实体的单个实例。它支持嵌套，允许您通过一对一或一对多关系访问实体。在您的实体中，任何数据都可以用**常见小组件**和**输入小组件**等类别中的一个小组件来显示。这是唯一能够显示通过上下文传递给页面的单一对象的小组件。它还能够监听其他小组件。

- **数据网格**

- 此小组件以表格方式显示数据，并且最常用于显示对象列表。它支持许多开箱即用的网格功能，并挂钩服务器端的排序、搜索和分页等功能。此小组件通常从数据库获取其数据，也可通过关联获取数据。

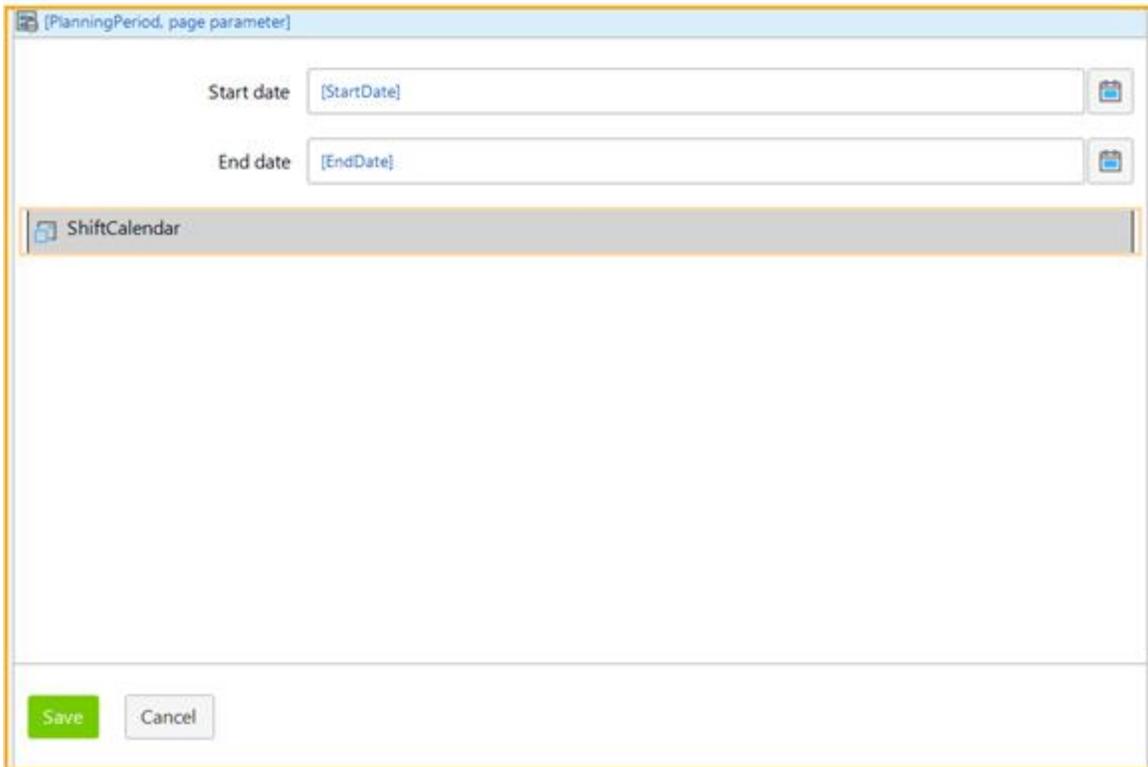
- **模板网格**

- 此小组件在图块视图中显示对象列表。例如，模板网格显示员工列表及个人资料图片。模板网格提供的控件可用于浏览、搜索和操控这些对象。如同**数据网格**一样，此小组件通常从数据库获取数据，也可使用关联。

- **列表视图**

- 此小组件显示对象列表。例如，您可以显示全体员工列表。每个对象都使用模板显示。将小组件放置在列表视图的拖放区中，可用于定义模板。此时，构建块派上用场了，它们为您在列表视图中构建单个项目打下很好的基础。如同**数据网格**和**模板网格**一样，此小组件通常直接访问数据库，也可使用关联。

- ☰ Data widgets
  - 📄 Data view
  - 📄 Data grid
  - 📄 **Template grid**
  - 📄 List view

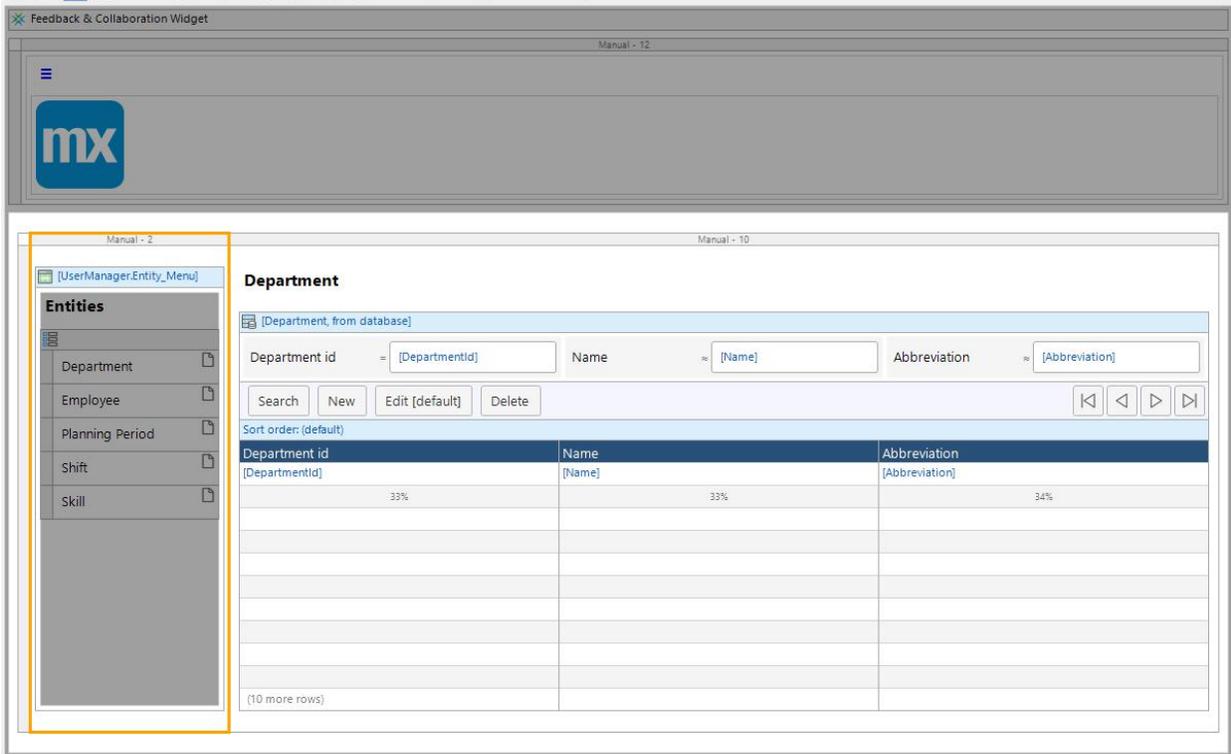


数据小组件可在工具箱组中找到，如图所示。数据视图（带蓝色标题）显示为示例，其中一些小组件与实体 **PlanningPeriod** 中的属性相联。在底部可看到两个操作按钮，用于保存数据或取消更改。

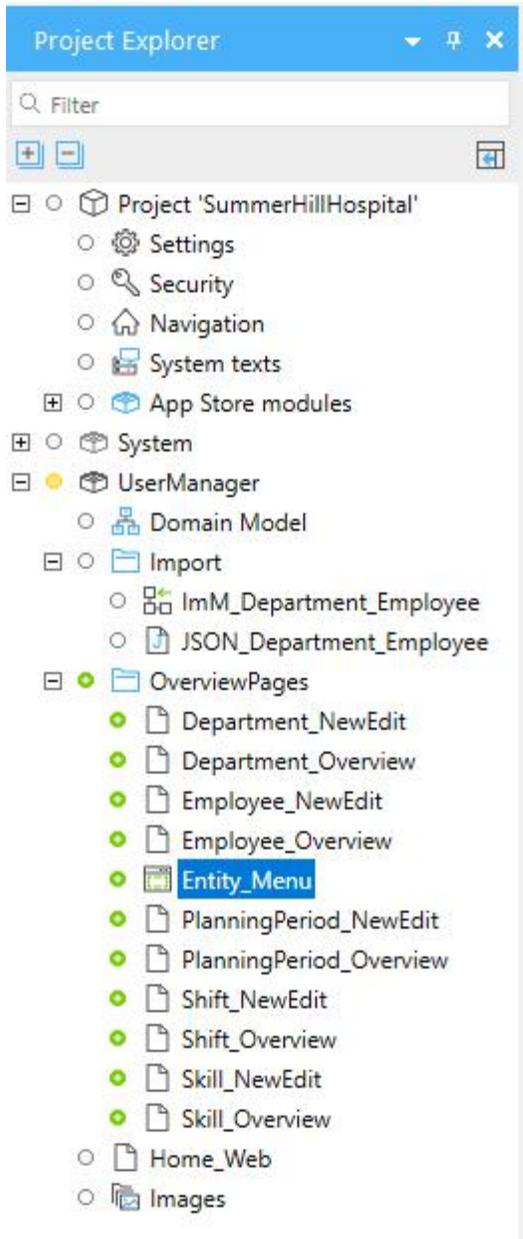
### 讲座 6.6.1：从概述页面移除导航菜单

Mendix 为您生成概述页面真的很棒，但其实不需要导航菜单，因为您将在整个项目中添加自己的导航。您可以使用所有概述页面中的片段调用来放弃该列。您还应将布局网格设置为自动填充。

1. 打开 **Department\_Overview**，删除有片段调用的列。



2. 选择带数据网格的列，然后双击以编辑列设置。将**桌面宽度**设置为**自动填充**，然后单击**确定**。
3. 对 **Employee\_Overview**、**Planning\_Overview**、**Shift\_Overview** 和 **Skill\_Overview** 页面重复这一步骤。
4. 从项目资源管理器中删除 **Entity\_Menu** 片段。



### 讲座 6.6.2: 调整员工页面

您现在有一个 Employee\_NewEdit 页面，而您希望为员工创建和编辑一个帐户，那么必须更新 Employee\_NewEdit 页面的布局。这里的关键是以一种巧妙的方式使用小组件。

Manual - 12

[Employee, page parameter]

Employee id [Employeeid]

Title [Title]

Firstname [Firstname]

Lastname [Lastname]

Gender [Gender]

Phone [Phone]

Jobtitle [Jobtitle]

Specialty [Specialty]

Email [Email]

Has account  Yes  No [HasAccount]

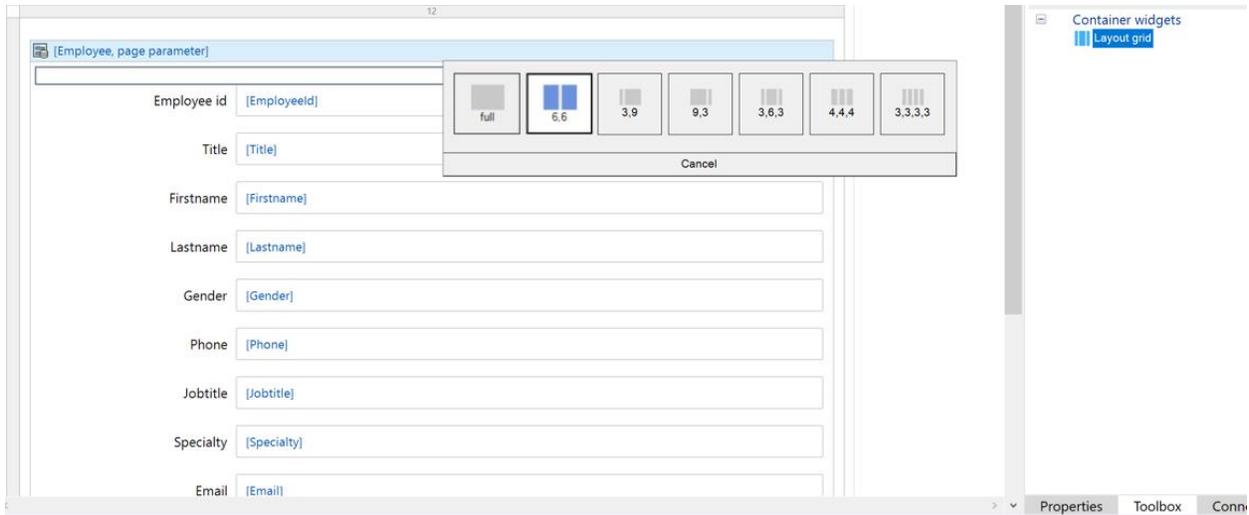
Department [Employee\_Department/Department/Name]  

Skill [Employee\_Skill/Skill/String]  

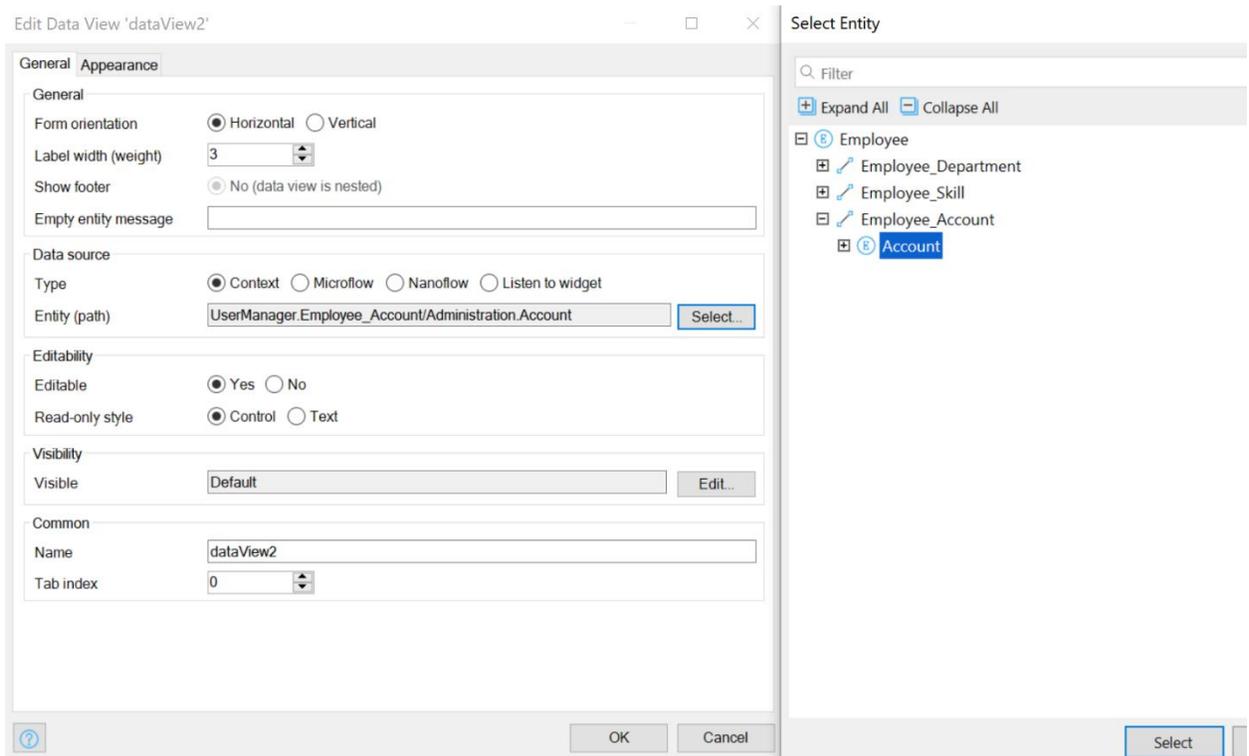
Account [Employee\_Account/Account/FullName]  

您的页面应像屏幕截图一样。

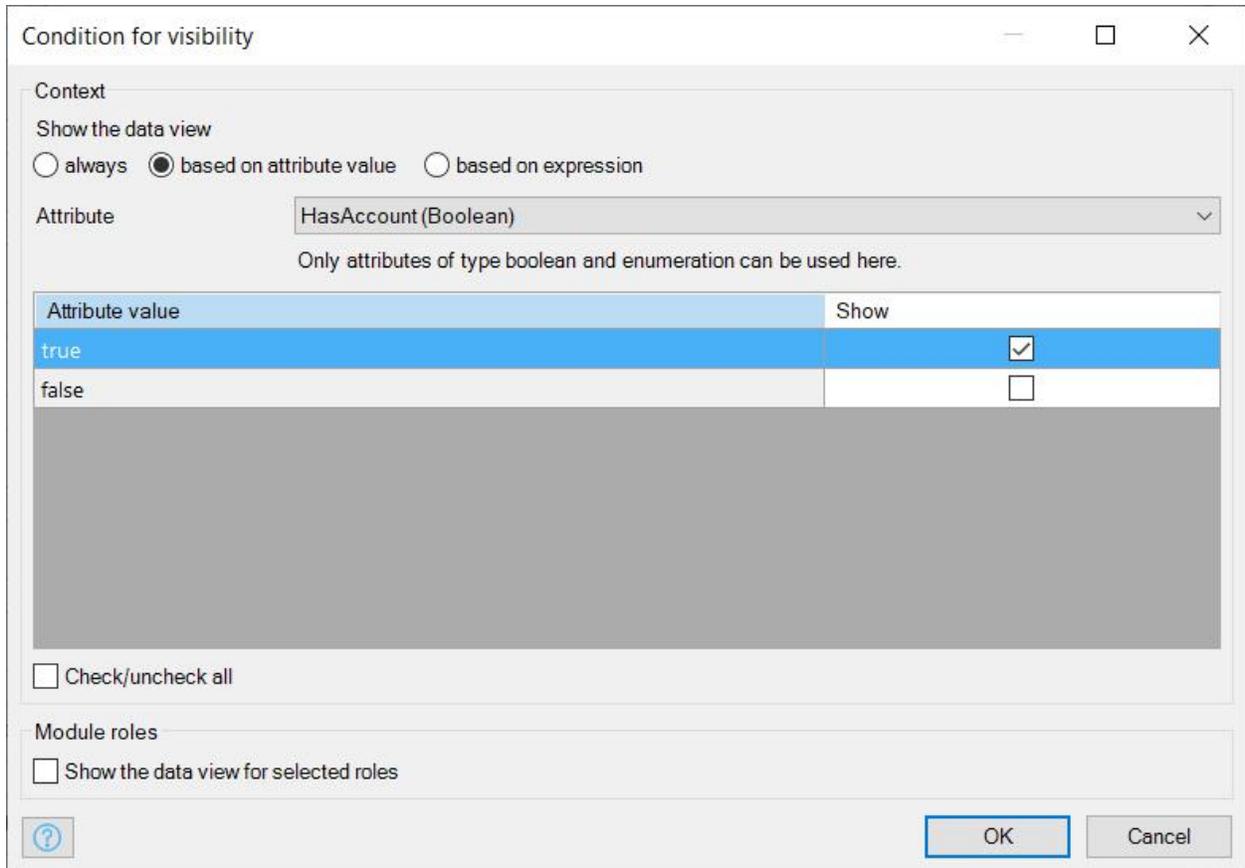
1. 在工具箱中，查找一个**布局网格**，然后将其拖到“员工 ID”小组件上方的“数据”网格。
2. 从出现的布局选项菜单中选择 **6,6**。



3. 将所有输入小组件拖到您刚刚添加的布局网格的左列中，然后删除帐户小组件 — 我们不需要它。
4. 将 **HasAccount** 复选框拖到右列中。
5. 在 **HasAccount** 复选框下面添加一个数据视图。单击它，将实体（路径）设置为关联 **Employee\_Account** 上的帐户实体。



1. 在可见性部分，单击屏幕右侧的编辑... 按钮。
2. 将显示数据视图选项设为基于属性值，然后选择属性 **HasAccount**。



- c. 单击**确定**。
6. 单击**确定**。
  7. 选择**是**自动填充**数据视图**。

现在，您的屏幕已准备好用于创建和编辑**帐户**实体。屏幕中有很多不需要的小组件，让我们在下一个讲座中解决这个问题。

### 讲座 6.6.3：编辑关联的帐户实体

1. 将**名称**的标签标题更改为**用户名**
2. 配置用户角色选择小组件
  1. 将数据源更改为**名称**，而不是 **ModelGUID**。
  2. 将**可编辑性**设置更改为**默认**。
  3. 单击**确定**。
  4. 右键单击用户角色小组件，并单击**选择选择页面**。
  5. 将**单击时**事件更改为**显示页面**。
  6. 使用 **PopupLayout** 和 **Select With List View** 模板，创建一个名为 **UserRole\_Select** 的新页面。Mendix 将为您创建这个页面。

3. 从工具箱获取“调用微流”按钮，并拖放到数据视图底部。
  1. 查找并选择微流 **Administration.ShowPasswordForm**。
  2. 将密码按钮的标题更改为**更改密码**。
  3. 将**按钮样式**设置为**警告**。
  4. 单击**外观**选项卡，将属性**自我对齐**设置为**右**。
  5. 操作操作完成后，单击**确定**。
4. 删除下图中未显示的任何字段。

The screenshot shows a data form titled "[Employee, page parameter]" with two main sections: "Auto-fill" and "Auto-fill".

**Left "Auto-fill" section:**

- Employee... [EmployeeId]
- Title [Title]
- Firstname [Firstname]
- Lastname [Lastname]
- Gender [Gender]
- Phone [Phone]
- Jobtitle [Jobtitle]
- Specialty [Specialty]
- Email [Email]
- Departme... [Employee\_Department/Department/]
- Skill [Employee\_Skill/Skill/String]

**Right "Auto-fill" section:**

- Has acco...  Yes  No [HasAccount]
- [Account, over association 'Employee\_Account']
- Name [Name]
- Last login [LastLogin]
- Blocked  Yes  No [Blocked]
- Active  Yes  No [Active]
- User role [UserRoles/UserRole/Name]
- Language [User\_Language/Language/Code]
- Time zone [User\_TimeZone/TimeZone/Code]
- Change Password** (highlighted in orange)

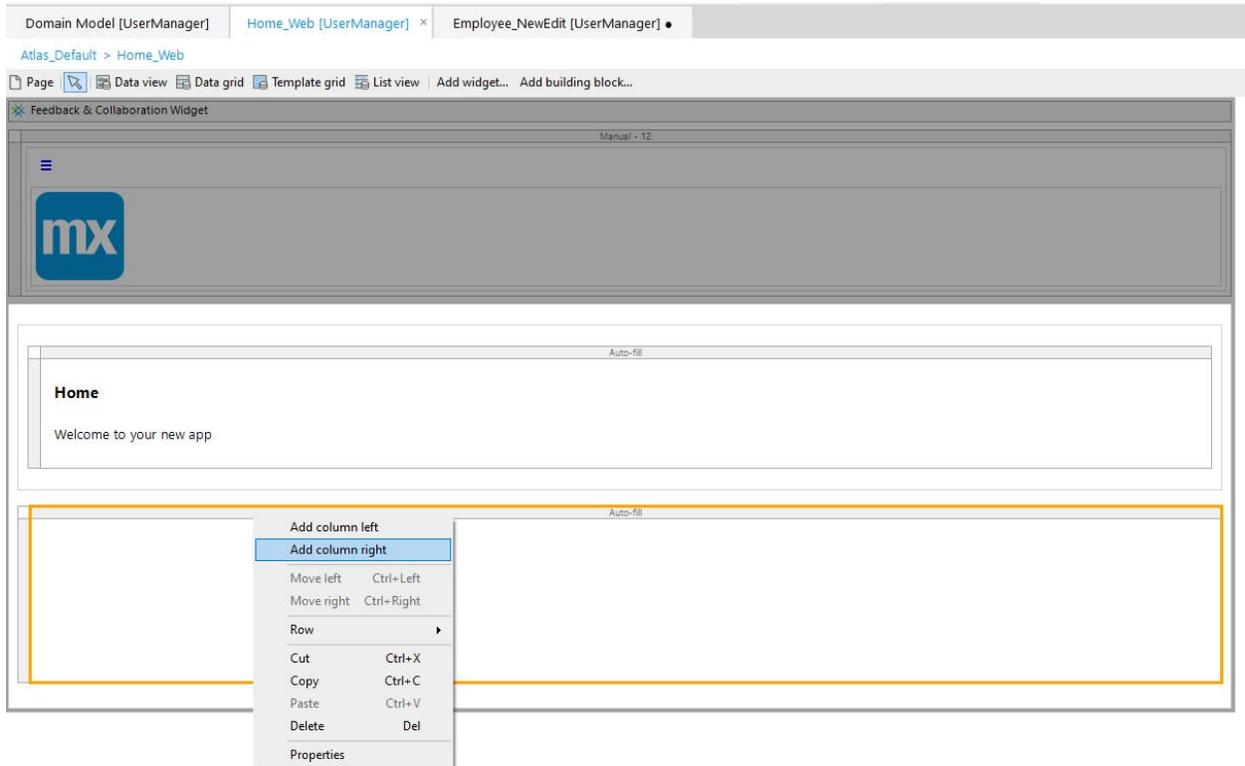
At the bottom, there are "Save" and "Cancel" buttons.

太棒了！您真正改进了该页面的设计。Summerhill 医院的人员会很满意您设计的完美页面并从中获得不小的便利。

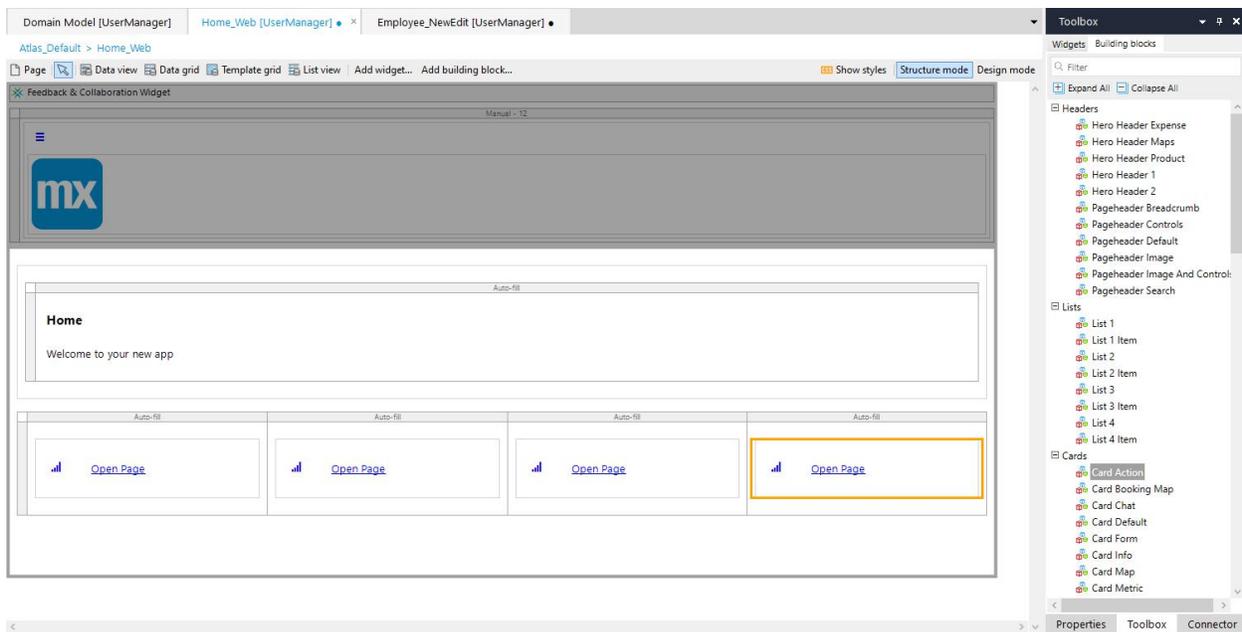
#### 讲座 6.6.4：设置主页

您已经拥有了自己的页面，现在是时候为它设置用户访问功能。实现此目标有多种方式，主页是最佳方式之一。这样，位于 **Summerhill** 医院的用户可以快速找到所需的信息。目前，添加 4 个按钮即可，这些按钮链接到部门、员工、技能及规划期间的概述页面。

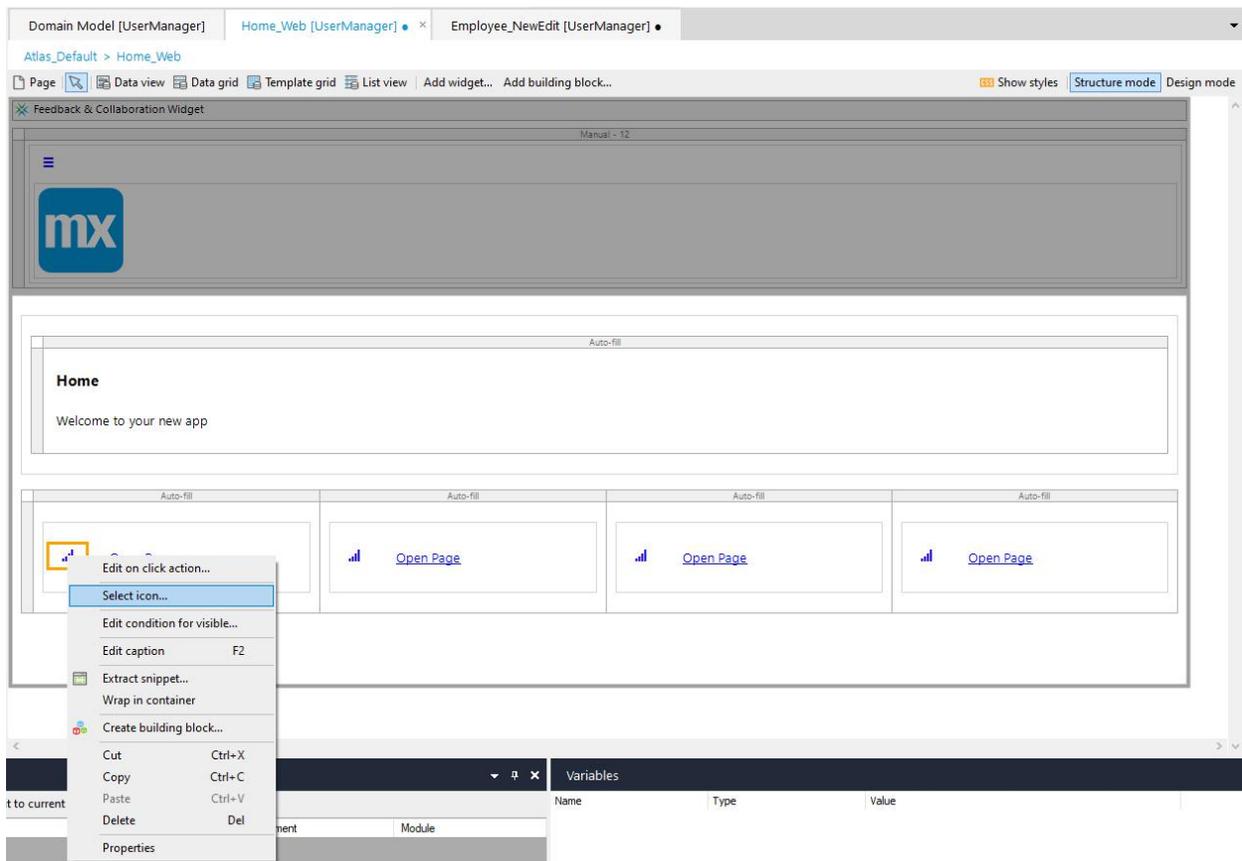
1. 打开 **Home\_Web** 页面，在底部布局网格中添加三列。



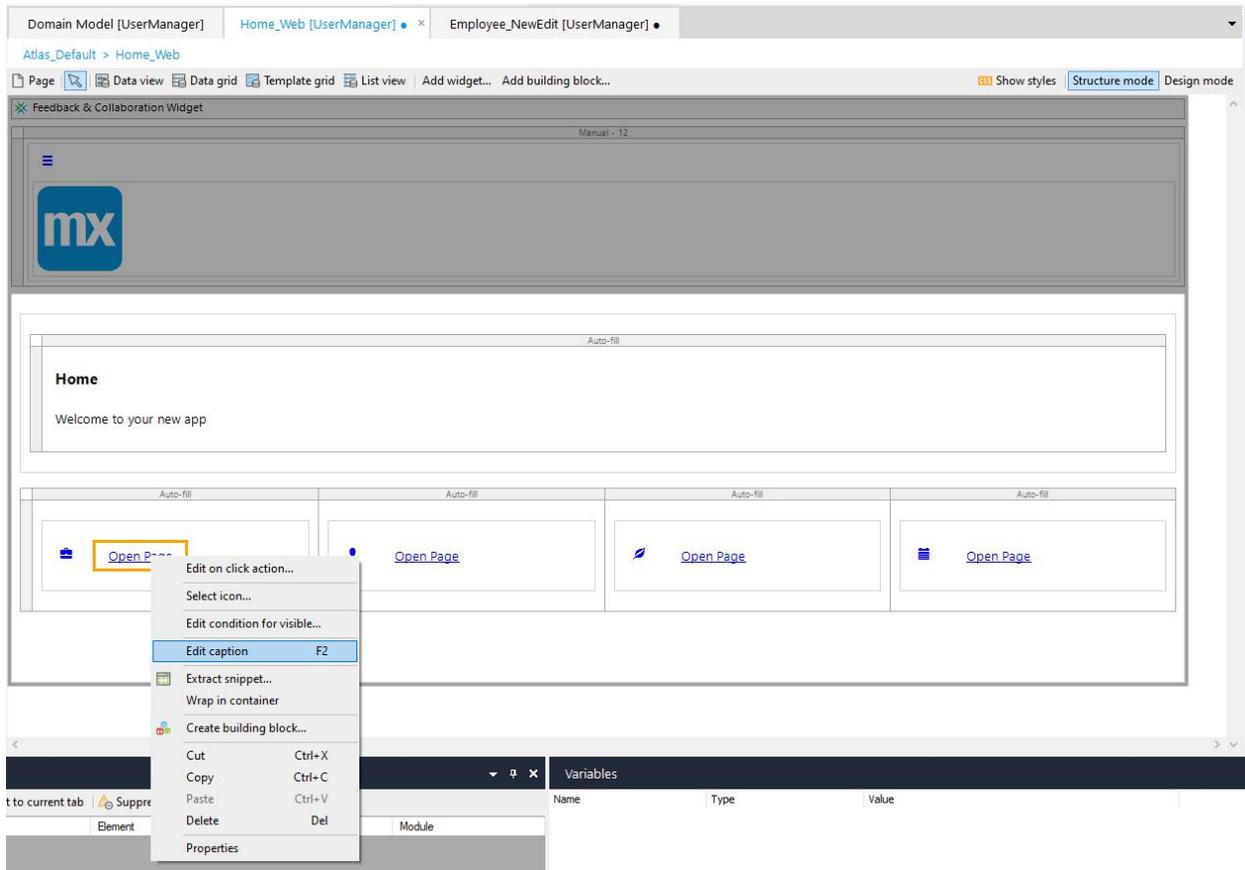
2. 在工具箱**构建块**选项卡中，将 4 个**卡操作**构建块添加到上一步中创建的每个列。



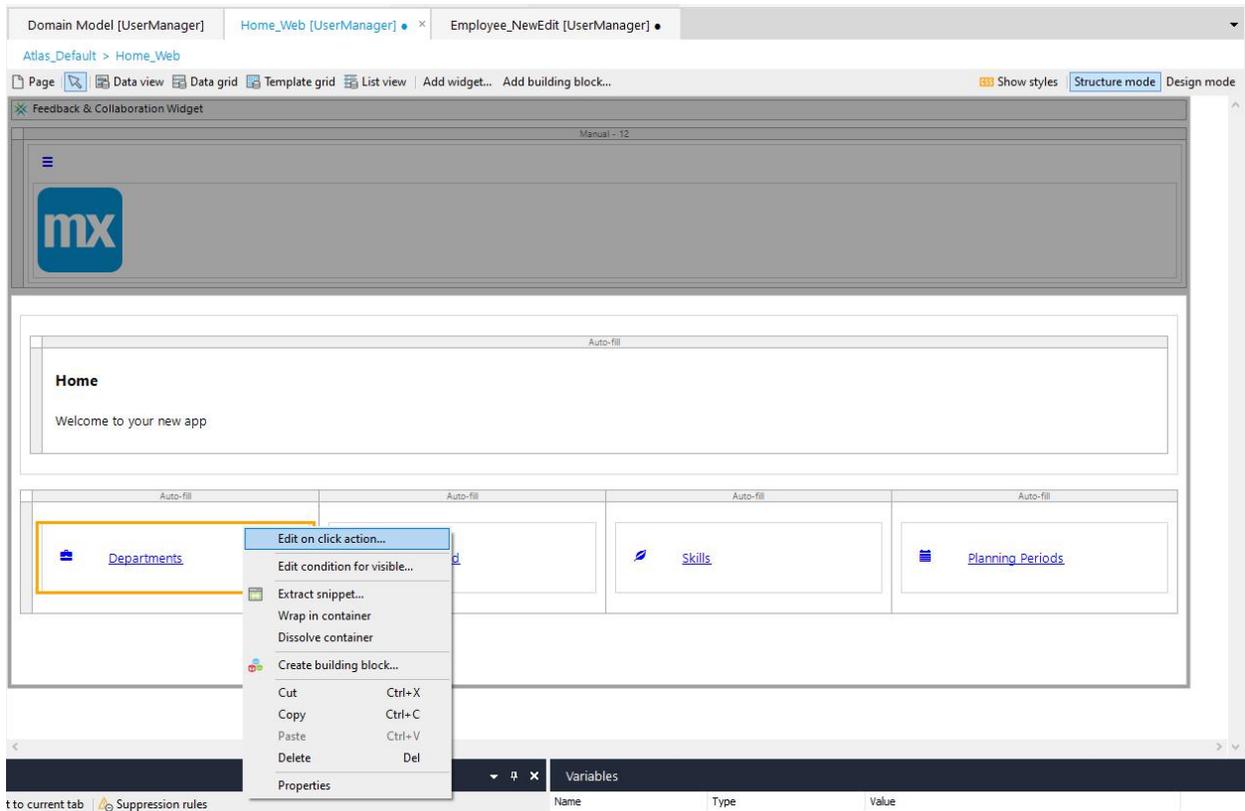
3. 将图标设置为公文包、用户、叶子和日历。



4. 将每个卡操作构建块的文本开放页分别改为部门、员工、技能和规划期间



5. 选择图标和文本周围的容器，右键单击并选择**编辑单击时操作**。设置每个按钮以链接到相应的概述页面。



好极了！您构建了一个出色的主页，使 Summerhil 医院的人更容易在一开始就上手使用该应用程序。下一课讲解导航菜单，即使不在主页上，用户也可以轻松地在页面之间切换。

## 讲座 6.7：导航菜单

Mendix 还配有导航菜单等功能。您可以直接从您的项目中编辑它。它位于项目资源管理器中的项目节点下，名为**导航**。单击它，您将看到一个包含两个选项卡的窗口。一个叫**响应式**，另一个叫**手机浏览器**。这些是**导航配置文件**。当客户端访问 Mendix 应用程序时，运行时将确定使用哪种类型的客户端，并基于客户端发送的 **User-Agent** 字符串为客户端选择适当的配置文件。

User-Agent String Regular Expression	Device Type
Android.*Mobile iPhone iPod BlackBerry	Phone
Android iPad	Tablet
(other)	Responsive

Mendix 提供的**导航配置文件**有四个：

- 响应式
- 平板电脑浏览器

- 手机浏览器
- 本地移动应用

**响应式**配置文件是默认设置，无法删除。如果没有合适的配置文件可用，**Mendix** 将使用此配置文件。

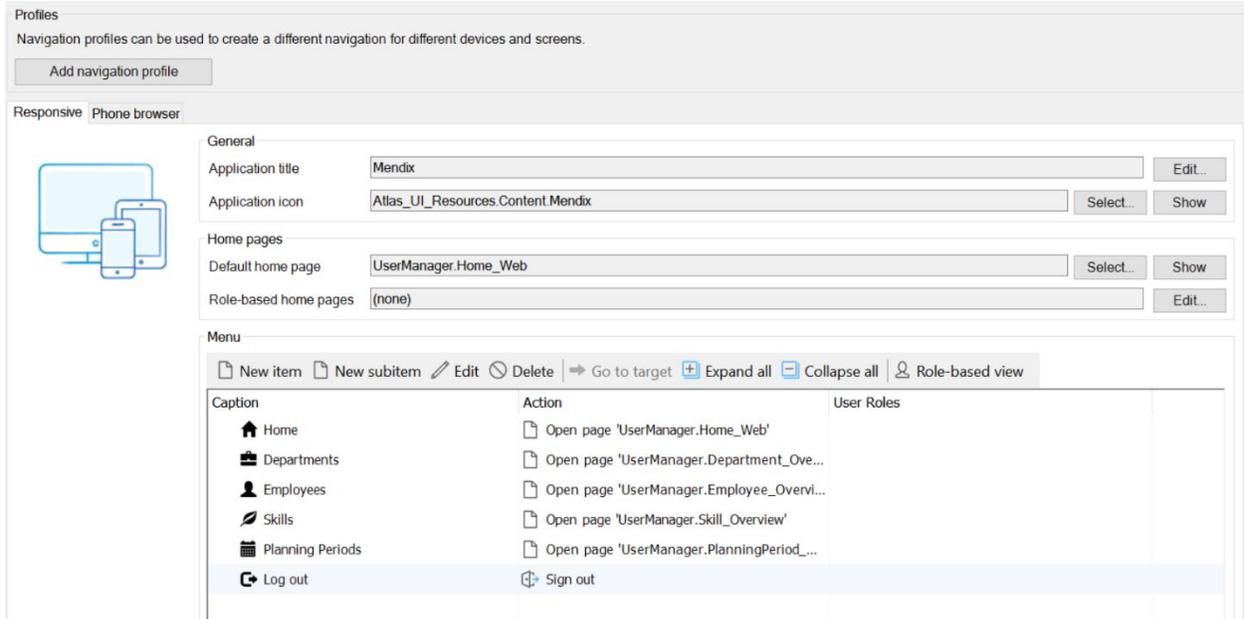
配置文件为应用程序设置标题，并决定哪个页面用作主页。此外，它还提供创建导航菜单的选项。您可以选择单击**新建项目**来创建菜单项目，并单击**新建子项目**来创建子菜单项目。每个项目都可以有一个标题、一个图标和一个操作。现在让我们将新学到的知识投入实践应用吧。

### 讲座 6.7.1：创建导航菜单

默认情况下，每个**导航配置文件**都有一个指向主页的链接。您想为您的概述页面添加项目。

1. 单击**新建项目**，并赋予其标题**部门**
2. 作为图标，单击窗口右侧的**选择...**来选择**公文包**
3. 设置**单击时**操作，以**显示页面**。
4. 选择 **Department\_Overview** 页面。
5. 针对以下实体重复此过程：
  1. 带用户图标的**员工**
  2. 带叶子图标的**技能**
  3. 带日历图标的**规划期间**
6. 添加最后一个项目，标题为**退出**，图标为**退出**。设置**单击时**操作以**签出**。

您的**响应式**导航配置文件现在应如下图所示。



导航菜单将帮助 Summerhill 医院的人们无论在哪个页面上都能轻松浏览应用程序，这都源自于您的辛勤付出。而您自己在整个实施过程中也受益匪浅。您一定会获得团队的好评认可！

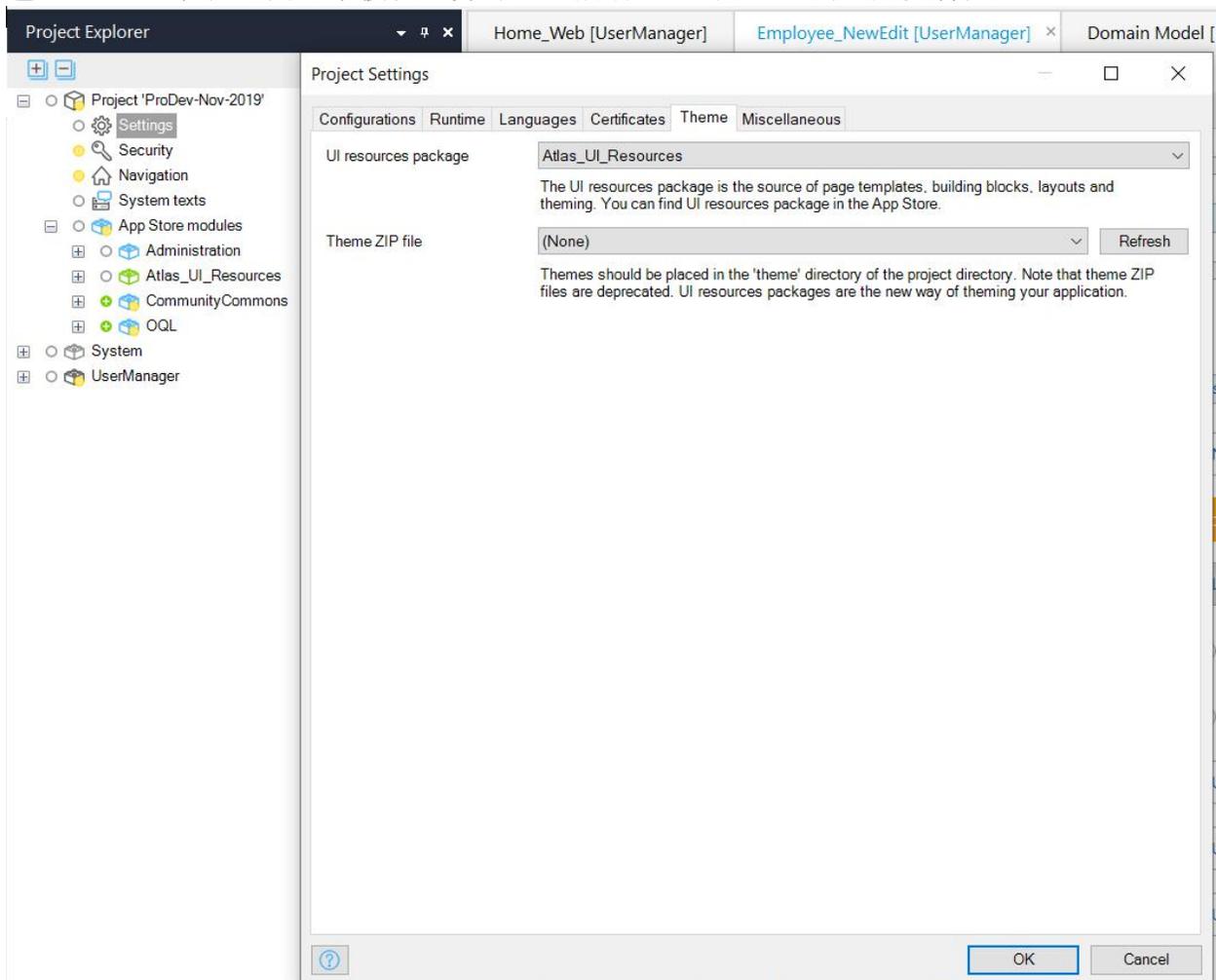
### 讲座 6.8: 设置页面样式

Mendix 提供很多开箱即用的主题设计，但即使是最好的主题设计也无法面面俱到。因此，**Mendix** 允许您定制设计主题。可直接在您项目的主题文件夹中使用 **Sass** 来执行此操作。除此之外，还提供一个 **github** 存储库，可作为您的个性化主题的基础。可在 **Mendix/ux-theming** 找到。



最后要提出的重要一点是，您可以创建自己的主题模块。如果打开项目节点并双击**设置**，会看到**项目设置**窗口中显示**主题**选项卡。无论选择哪个模块作为**用户界面资源包**，都将应用于所有的主题设计。特别注意，这也包括所有的页面模板和布局。最好是以

**Atlas\_UI\_Resources** 模块为基础开始。完成后，您可以导出此模块并送给同事、客户或上传到市场，以便其他人从您的主题设计技能中获益。鉴于您将该模块设为**用户界面资源包**，Mendix 默认连同这个模块一并导出您所有 Sass 和 CSS 的主题文件夹。



### 讲座 6.8.1: 添加定制样式

开箱即用的样式通常就足以满足要求了。但有时候您想改变一下样式。您的最终客户可能期望呈现完全不同的外观，或者某些默认样式不适用于您的项目。在此特定情况下，应用程序中的弹出窗口相对较小。这对于大多数用户来说并不是什么大问题，但由于您对 **Employee\_NewEdit** 页面的布局做了相当大的改动，弹出页面的宽度就成了一个问题。因此，我们现在来着手更改此应用程序的样式。

1. 从本网站下载 **Visual Studio Code** 并安装它。
2. 从 [Mendix 文档网站](#) 下载样式应用程序 **Calypso** 并安装它。
3. 将**应用程序集合**设置为包含您的 Mendix 项目的文件夹。
4. **应用程序文件夹**下拉框中包含文件夹内所有 Mendix 项目的列表。请注意，它并不包含子文件夹中的 Mendix 项目。选择 **SummerHill 医院** 项目。
5. 单击**开始**，开始对您项目中的主题文件夹进行观察。

6. 在 Visual Studio Code 中打开项目文件夹。

在**主题**文件夹中，您可以找到许多文件。在这种情况下，您想要在**样式**文件夹中查找，因为这将包含所有相关文件。要调整网页的样式，您应在 **Web** 文件夹中查找。**本机**文件夹包含本机应用程序的样式。

**Web** 文件夹包含一个 **css** 文件夹和一个 **sass** 文件夹。**css** 文件夹包含将移至部署文件夹的文件，不可直接编辑此文件夹中的文件，**Calypso** 将为您管理此文件夹的内容。

**sass** 文件夹是设置样式的地方，尤其是**应用程序**文件夹。在任何其他文件夹中添加样式不会破坏您的样式设计，但难以跟踪您的更改。因此，建议在其中一个 **\_custom** 文件中进行小的更改，或者创建您自己的文件并在 **\_custom.scss** 中导入。

7. 前往**主题** → **样式** → **网页** → **sass** → **app**，打开 **\_custom.scss** 文件。

8. 在 **@import "custom-variables"** 的下方添加以下 SASS 样式；确保弹出页面的大小足以容纳您添加的新字段。

```
.mx-window {  
  width: 1200px;  
}
```

很好，您的应用程序现在看上去很不错，您也了解到许多有关如何使用 Mendix 构建应用程序的新知识。

## 总结

有大量信息要消化，更有无限可能等待您的发现和探索。目前您已大致了解 **Mendix 用户界面功能**。以丰富充足的知识储备为基础，让我们一同进入下一个精妙主题：使用逻辑与数据进行交互。