07 - 模块 7：数据交互

模块 7：数据交互

模块简介（模块图块）：在此模块中，您将学习如何使用微流和纳米流进行数据交互。

讲座 7.1：简介
您在短时间内构建了许多功能，但到目前为止，您只用到了 Mendix 默认功能。尽管这样做对简单设置很管用，但有时您确实需要一些定制逻辑从 HR REST 定期检索信息。数字化创新团队要求每天至少提取一次数据。他们还希望有一项按钮之类的设置，可在急需新数据时触发同步。将所有用户拉入系统后，他们需要能够区分本地创建的用户和导入的用户。Mendix 为您达成所愿，只需利用一些微流即可做到。

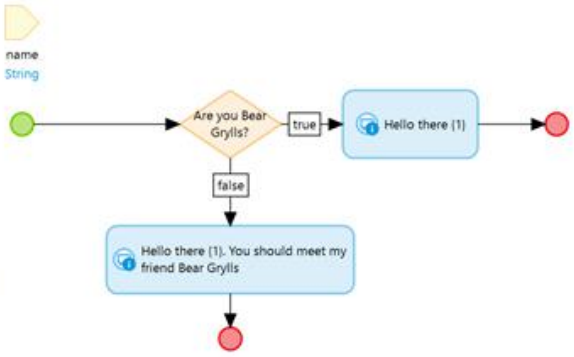
在此模块中，您将了解以下内容：

- 什么是微流？
- 微流的命名规则。
- 微流中的哪些元素可以为您所用。
- 哪些触发器可触发微流。
- 如何与微流内的数据进行交互。
- 如何调试微流。

现在我们开始探索定制逻辑在 Mendix 中的运作方式！

讲座 7.2：什么是微流？
微流是 Mendix 用于创建定制逻辑的方式。它是一种表示代码的直观方式。

```
1   function SayHello(name)
2   {
3       if(name == "Bear Grylls")
4       {
5           displayMessage("Hello there " + name);
6       }
7       else
8       {
9           displayMessage("Hello there " + name +
10              ". You should meet my friend Bear");
11      }
12  }
```



代码和微流之间的显著差异在于，微流基于**活动**。这些是图像中的蓝色矩形。您可以把这些活动比作指令，但在某些情况下，它们更接近于功能。其他核心组件包括**拆分**（对应项为**合并**）、**事件**、**参数**、**循环**和**调用其他微流**。您还会发现处理数据的活动，告诉

运行时执行一个操作（例如从外部服务中检索数据），或指示客户端执行一个操作（例如打开页面、关闭页面或下载文件）。

通过这些核心组件，Mendix 提供了一种图灵完备的语言，您可以用它来表达自我。它允许您将一系列操作串连到逻辑流中。然后将此流存储在 Mendix 应用程序中，一次执行一个活动。

微流由运行时执行。执行开始后，运行时启动数据库事务。在微流结束时，事务提交。如果在任何点遇到错误，整个微流将回滚。如果微流包含对子微流的调用，运行时会在子微流的起点和终点设置保存点。这意味着所有子微流将在根微流事务中运行。

如果默认提供的操作不够用，您可以随时转向市场，寻找包含额外操作的模块。这些操作在 Java 中实施，您可以将模块添加到您的项目中。该模块包含这些操作的完整 Java 源代码。这些操作使用与内置操作相同的界面，因此 Mendix 编写的活动与您从市场下载的活动之间没有差别。如果没有任何模块提供所需的操作，您始终可以使用 Java 编写自己的操作。这些 Java 操作源文件随您的项目一起存储，并在您部署应用程序时编译。

在图像中，您可以看到伪代码与微流之间的对比。通过代码进行微流的好处是，您可以更轻松地与非技术人员讨论微流。他们不需要处理的任何复杂性都可以隐藏在子微流或 Java 操作中。

讲座 7.3：命名规则
正如在其他编程语言中一样，您希望赋予微流一个描述性名称。Mendix 规则包括触发微流的事件类型、正在处理的主实体的名称以及正在执行的工序。此规则可生成以下模板：

{Prefix}_{Entity}_{Operation}。

存在一些例外情况，例如，没有主实体，或者有充分理由使用不同的名称来提高可理解性。重要的是要确保微流名称清楚地表明其目的。

要轻松查找和识别微流的用途，您可以使用标准前缀。随着本模块的推进，我们将介绍您需要用到的前缀。要是您感兴趣的话，可查阅 Mendix 文档提供微流的标准前缀列表。

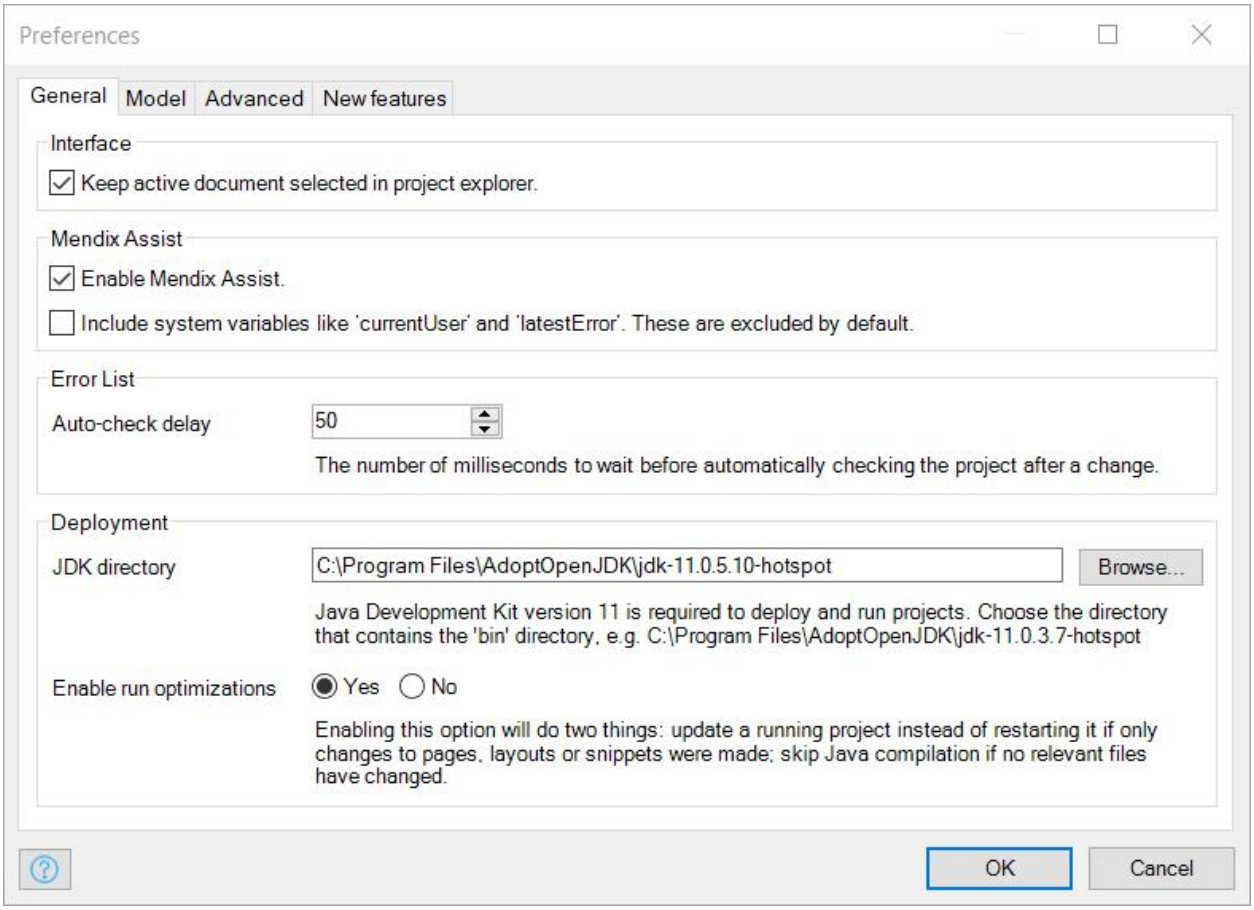
https://docs.mendix.com/howto/general/dev-best-practices#3-4-microflows

讲座 7.3.1：从 HR REST 服务检索员工和部门
您已经了解微流的外观以及如何命名它们，现在即可创建您的第一个微流。我们从简单任务入手，先执行一个活动。

1. 在**项目资源管理器**下，右键单击**导入**文件夹，单击**添加微流**，并将其命名为 **ACT_Department_Employee_ImportFromRest**

您看到蓝点了吗？这就是 Mendix Assist，它帮助您建立微流的人工智能。现在把它关闭。如果您没有看到，那也没关系。无论如何，我们总会找到设置的位置。

2. 单击**编辑** > **首选项...**，然后取消选中**启用 Mendix Assist** 旁边的复选框。如有需要，您随时都可以重新开启它，只需回逆之前的步骤即可。



3. 查看新的微流，前往**工具箱**，查找**调用 REST 服务**活动，然后将其拖到您的微流上。
4. 双击活动，在**常规**选项卡中将**位置**设置为：http://localhost:8079/rest/hospitalhr/v1/department。注意，如果更改了 MockHrService 的端口号，此 URL 可能会不同。请参考 **MockHrService** 中 **REST** 文件夹中的 **HospitalHr** 文档，找到正确的 URL。
5. 前往 **HTTP 标头**选项卡，并添加一个**定制标头**，其中的**键**设置为“接受”，**值**设置为'application/json'。考虑到此为字符串，必须加上单引号。
6. 转到**响应**选项卡，将**响应处理**设置为**应用导入映射**。
    1. 将**导入映射**设置为 **ImM_Department_Employee**。

2. 务必将**提交**设置为**是**，而不是**没有事件的“是”**。这一点将对后面的操作相当重要。

7. 为了导入数据，您希望将新项目添加到导航菜单以触发此微流。您可以为这个导航菜单项目使用**导入**图标。

非常好，这是您的第一个微流。让我们仔细查看用于构建微流的各个元素。

讲座 7.4：微流及相关元素

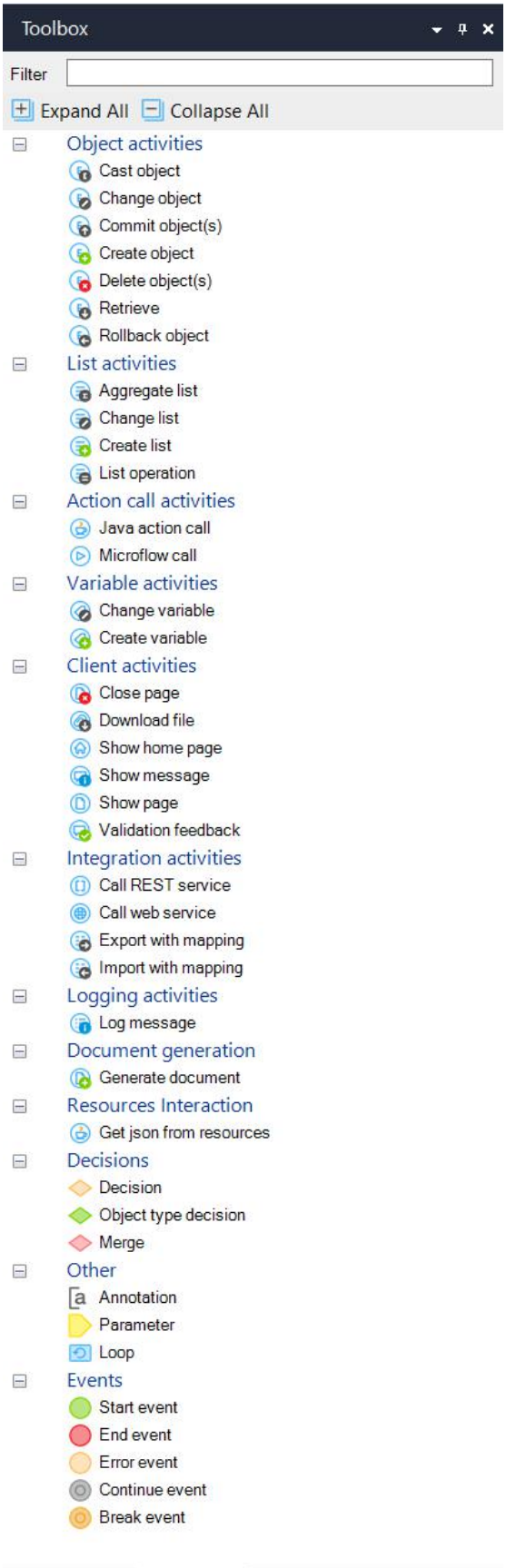
创建微流时，工具箱将提供以下元素。所有微流必须恰好有一个**启动事件**。您可以把它比作函数的开始部分。**结束事件**终止执行微流，可按需拥有多个事件，正如在函数中可按需返回多个语句一样。**序列流**表明哪些活动相互关联。**活动**是您的微流核心，可以把它比作您最喜欢的编程语言的一条指令或标准库中的函数调用。**决策**允许您对序列流进行分支，或将两个流重新合并在一起。**注释流**提供一种将**注释**或评论连接到活动的方法。它允许您直接在微流中编写文档。

Element	Type	Graphic
Start Event	Event	
End Event	Event	
Sequence Flow	Flow	
Activity	Activity	
Decision	Decision	
Merge	Decision	
Parameter	Artifact	
Annotation	Artifact	
Annotation Flow	Flow	

微流的触发方式有多种：

- 通过单击按钮的操作
- 通过用户界面元素上的进入/更改/离开事件
- 通过动态用户界面元素的加载事件
- 通过实体上的事件处理程序
- 通过项目导航菜单
- 通过定制导航菜单
- 通过导入映射
- 通过导出服务定义
- 通过时间表
- 您的应用程序启动后
- 您的应用程序关闭前
- 在运行状况检查期间（此触发器定期运行，可确保应用程序正常运行）

工具箱会预先加载活动列表。它们已分类，以便您轻松找到所需。除了分类之外，您还可以使用过滤器进行搜索。添加包含新微流活动的模块时，您将看到一个或多个其他的类别。

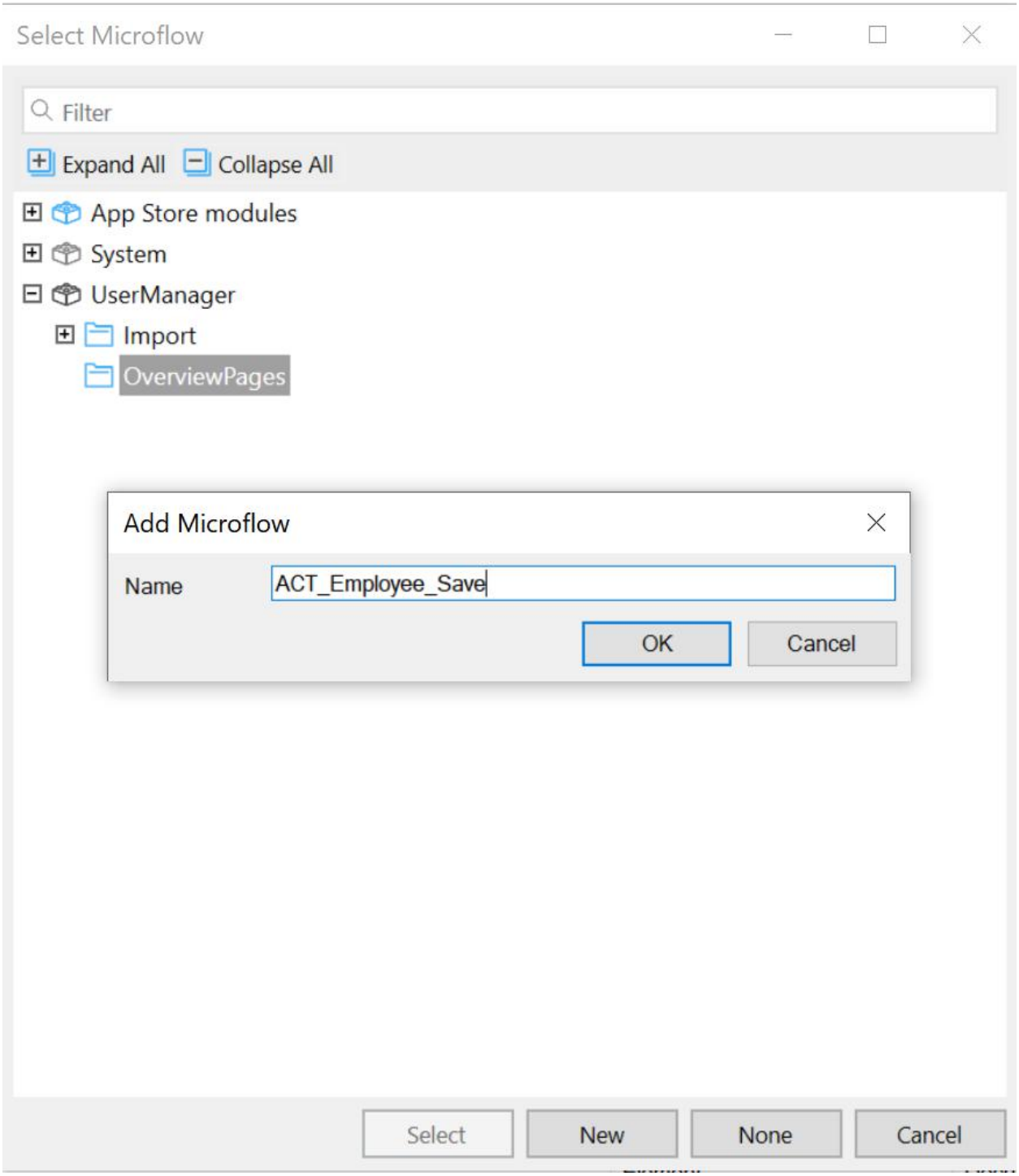

Toolbox
Filter
Expand All
Collapse All
Object activities
Cast object
Change object
Commit object(s)
Create object
Delete object(s)
Retrieve
Rollback object
List activities
Aggregate list
Change list
Create list
List operation
Action call activities
Java action call
Microflow call
Variable activities
Change variable
Create variable
Client activities
Close page
Download file
Show home page
Show message
Show page
Validation feedback
Integration activities
Call REST service
Call web service
Export with mapping
Import with mapping
Logging activities
Log message
Document generation
Generate document
Resources Interaction
Get json from resources
Decisions
Decision
Object type decision
Merge
Other
Annotation
Parameter
Loop
Events
Start event
End event
Error event
Continue event
Break event

熟悉这些元素的最佳方法就是投入使用，下面我们来看一下为 Summerhill 医院实施第一个微流。

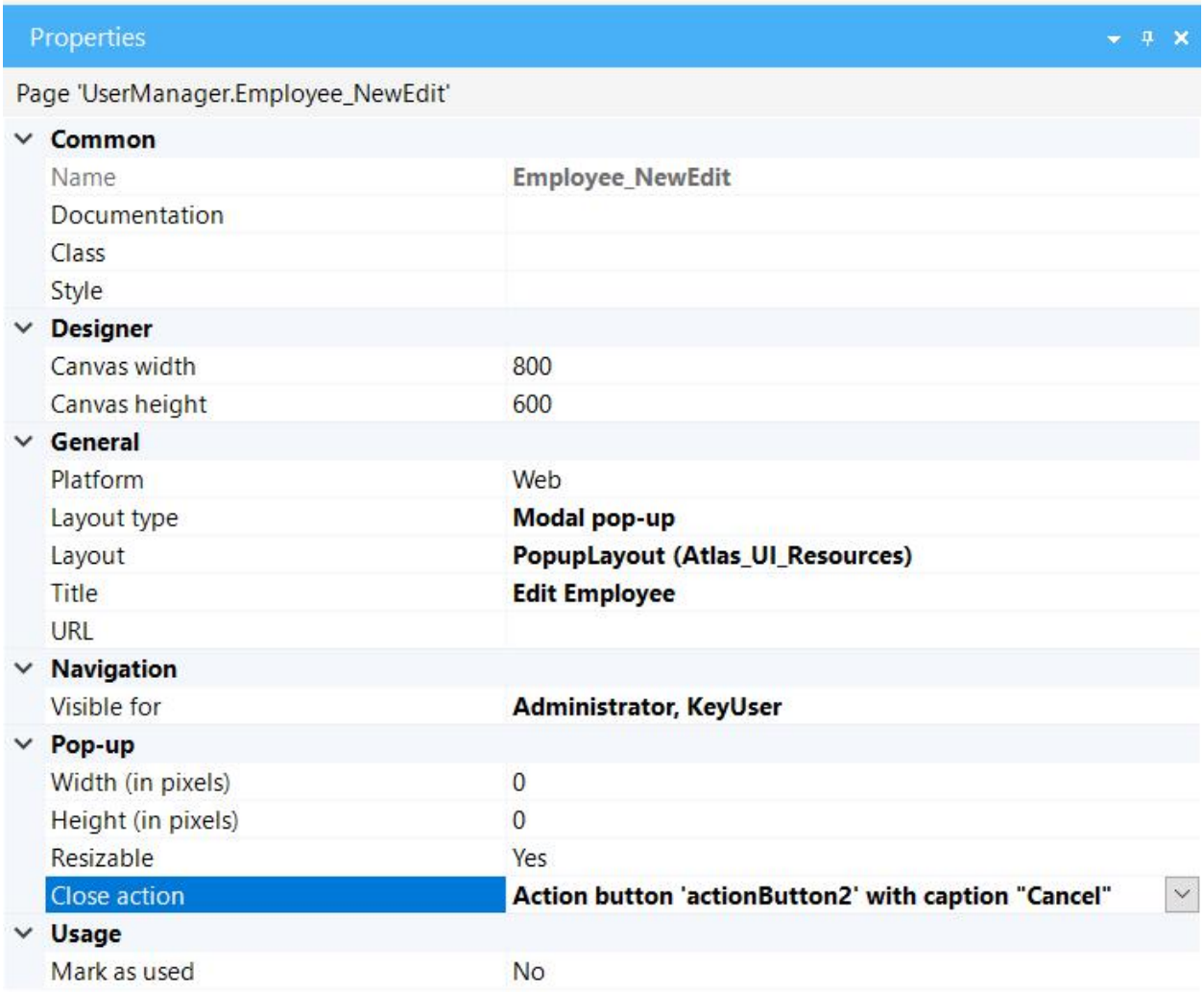
讲座 7.4.1：构建您自己的“保存”和“取消”微流

您已经向 **Employee_NewEdit** 页面添加了数据视图，并添加了与员工相关的实体，现在需要定制逻辑来确保妥善存储任何更改，或如果用户选择取消更改，那么会执行正确回滚。为此，您需要两个新微流，一个用于“保存”按钮，另一个用于“取消”按钮。让我们现在就开始吧！

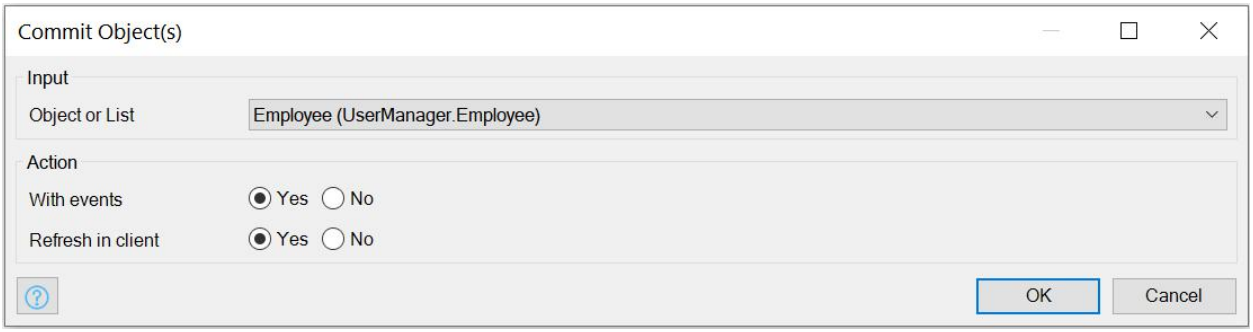
1. 打开 **Employee_NewEdit** 页面，右键单击**保存**按钮。选择**编辑单击操作**，并设定**单击时**操作以**调用微流**。
2. 单击**新建**以创建新的微流，并指定名称为 **ACT_Employee_Save**。



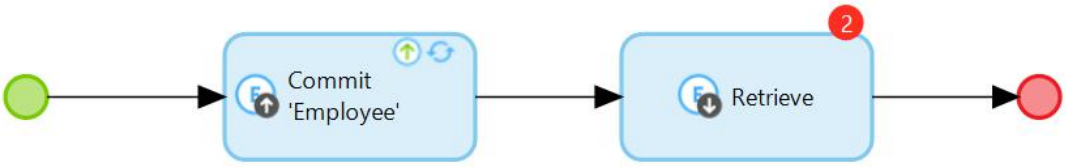
3. 单击**确定**。
4. 对**取消**按钮重复这一过程，并将该微流命名为 **ACT_Employee_Cancel**。
5. 弹出页面提供**关闭操作**选项，可用于配置当用户使用屏幕右上角的 **X** 关闭页面时发生的情况。要配置此功能，单击页面编辑器中页面外空白区域上的某个位置，确保您的属性窗格显示页面属性。将**关闭操作**设置为**取消**按钮。



6. 从**项目资源管理器中**打开 **ACT_Employee_Save** 微流程，并添加一个**“提交”对象**活动来提交**员工**实体。这一点是有必要的，因为在服务器上运行的微流会改变该实体。必须让客户认识到这一点。操作操作完成后，单击**确定**。



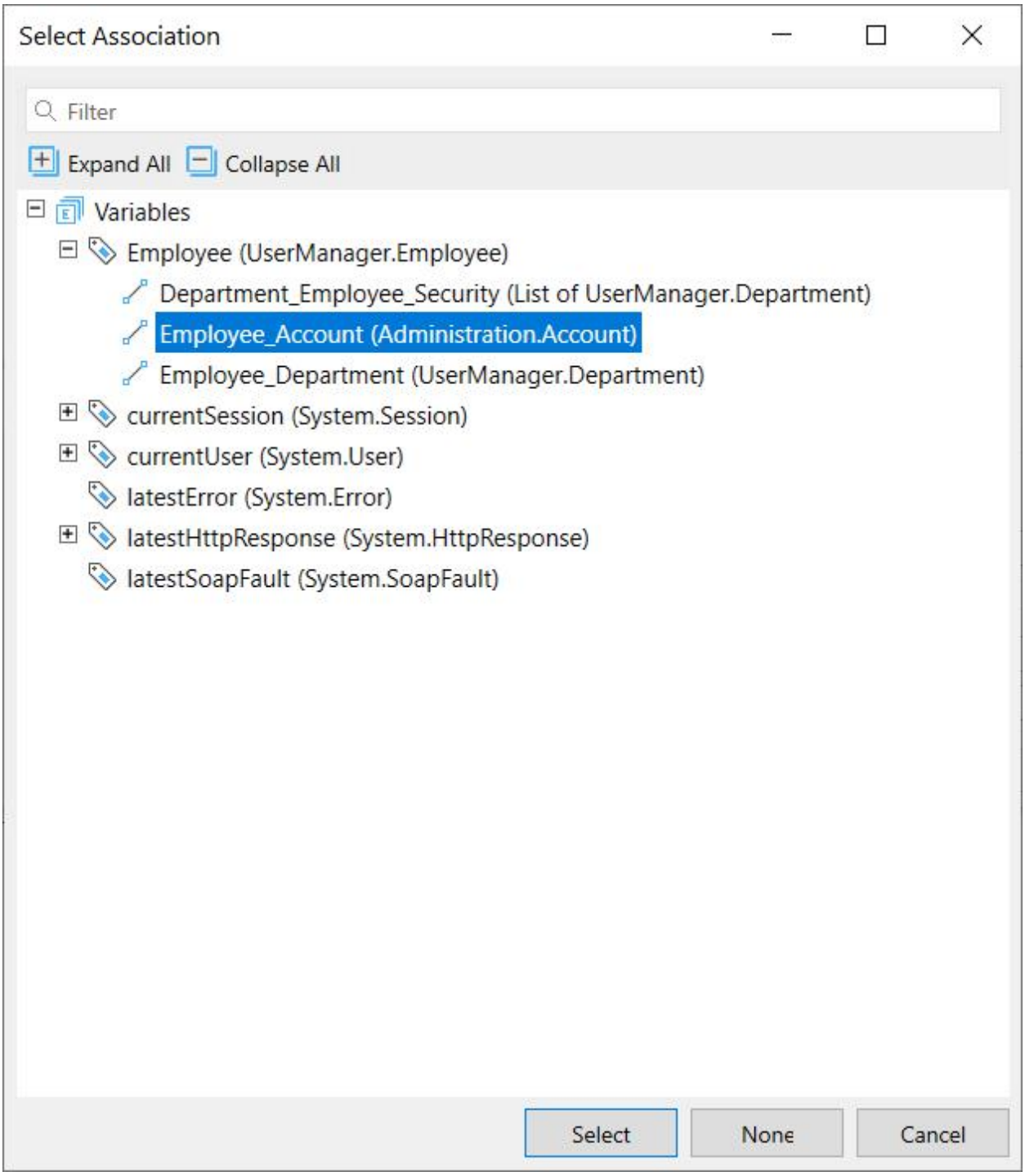
7. 为了能够提交**帐户**实体，您需要执行检索。为此，请前往**工具箱**，把一个**检索**活动拖到您的流程中，在您的**提交**活动右边，然后双击它。



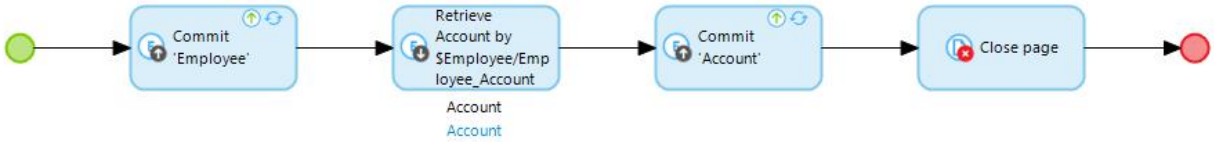
8. **检索**活动不知道要获取什么以及从哪里获取，所以您也需要配置该活动。双击**检索**活动并点击窗口右侧的**选择**按钮即可完成。

在 Mendix 中检索数据时，您有两个选项。可按关联进行检索，这样即可访问与范围内任何对象直接连接的所有实体。这包括尚未出现在数据库中的任何对象。若选择从数据库进行检索，即可在一次检索中使用多个关联来获得您需要的实体，但必须要知道的是，您无法检索到尚未提交或刚刚被删除的对象。

9. 在新打开的窗口中，单击**员工**实体旁边的加号图标，选择 **Employee_Account** 关联。单击**选择**，然后**确定**。



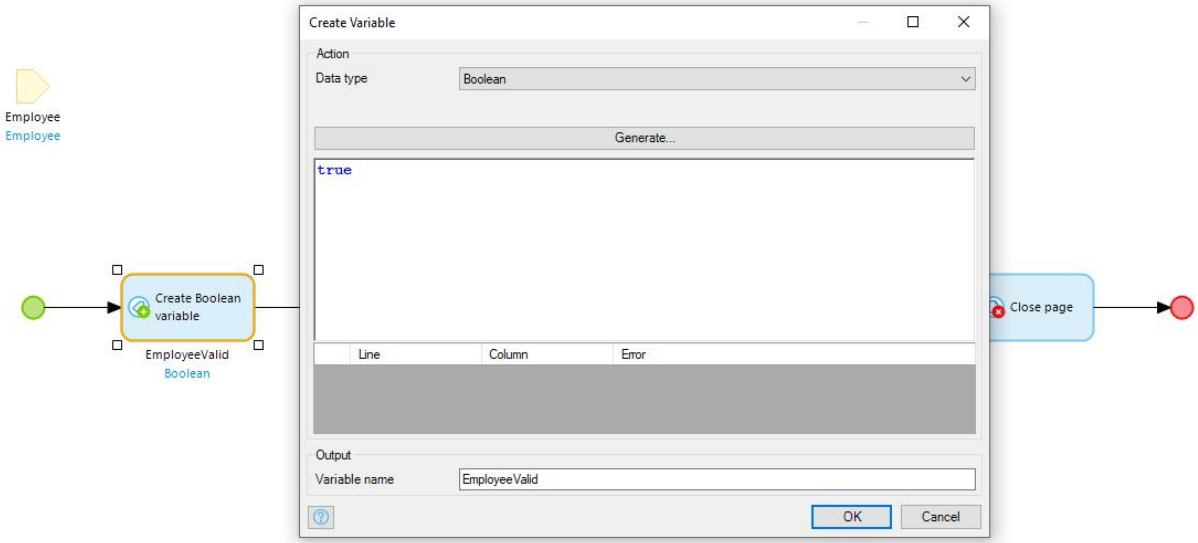
10. 将另一个**提交**活动拖到您的流程中，放在**检索**活动的右边，并对其进行设置以提交**帐户**对象。务必将**客户端刷新**设置为**是**。
11. 还有微流中的最后一个活动，就在第二个**提交**活动的右侧，即**关闭页面**活动。请继续添加。生成的微流如下所示。



讲座 7.4.2：添加验证规则

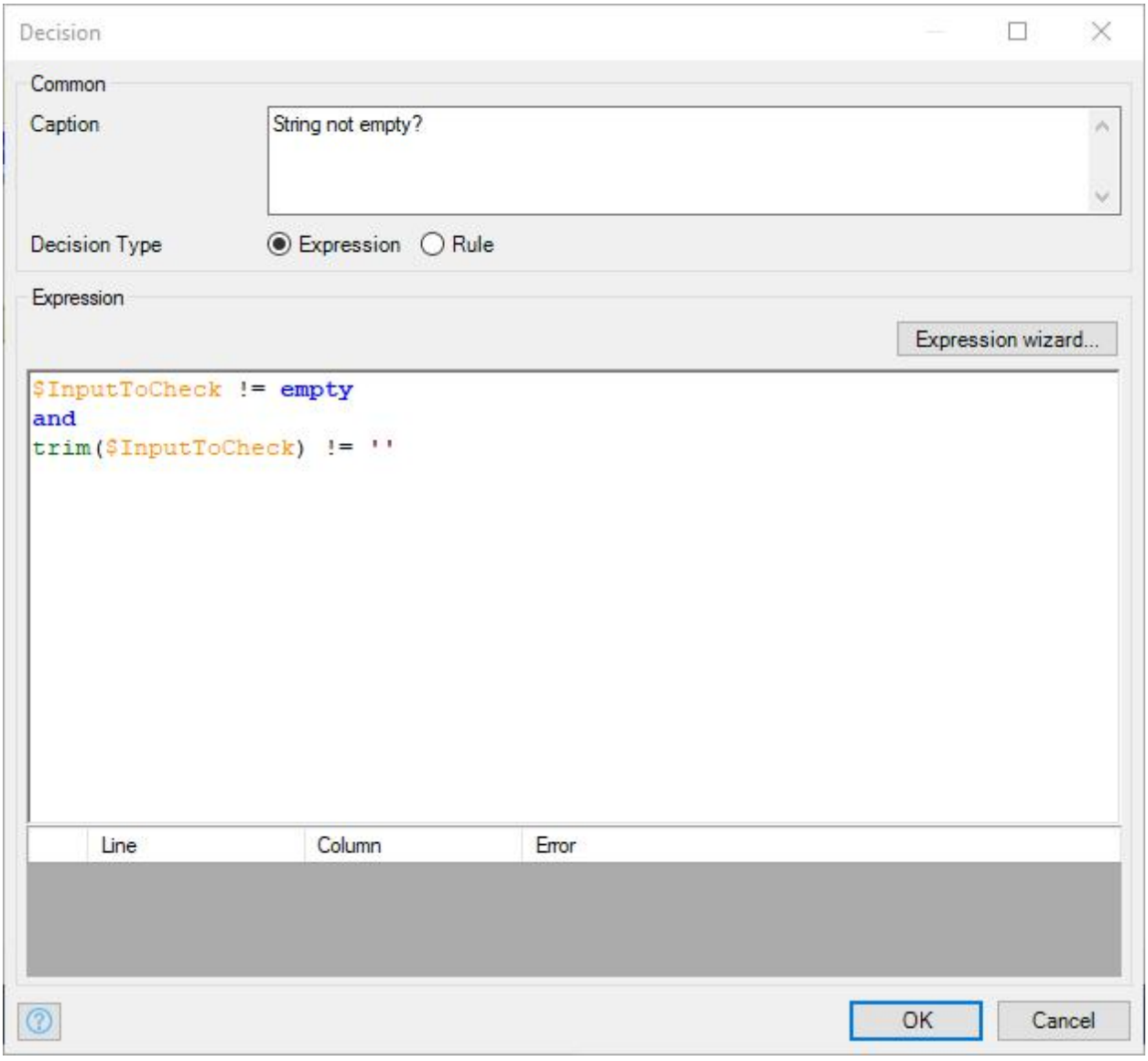
我们已经完成了基本设置，现在是时候添加一些验证了。在域模型中添加**验证规则**、使用**页面验证**的输入小组件或使用像您刚刚创建的微流均可做到这一点。最后一种方法就是您将使用的方法。这样做的原因是，如果使用**验证规则**，导入可能会失败。添加为**页面验证**可能会妨碍您轻松重新使用验证逻辑。此外，使用**验证规则**无法检查是否设置了某个关联。 因此，最好使用**微流验证**。

1. 在微流的开始部分添加一个**创建变量**活动。将**数据类型**设置为**布尔**型，并将值设置为 **true**。在**输出**部分中，将**变量名称**设置为 **EmployeeValid**。操作操作完成后，单击**确定**。

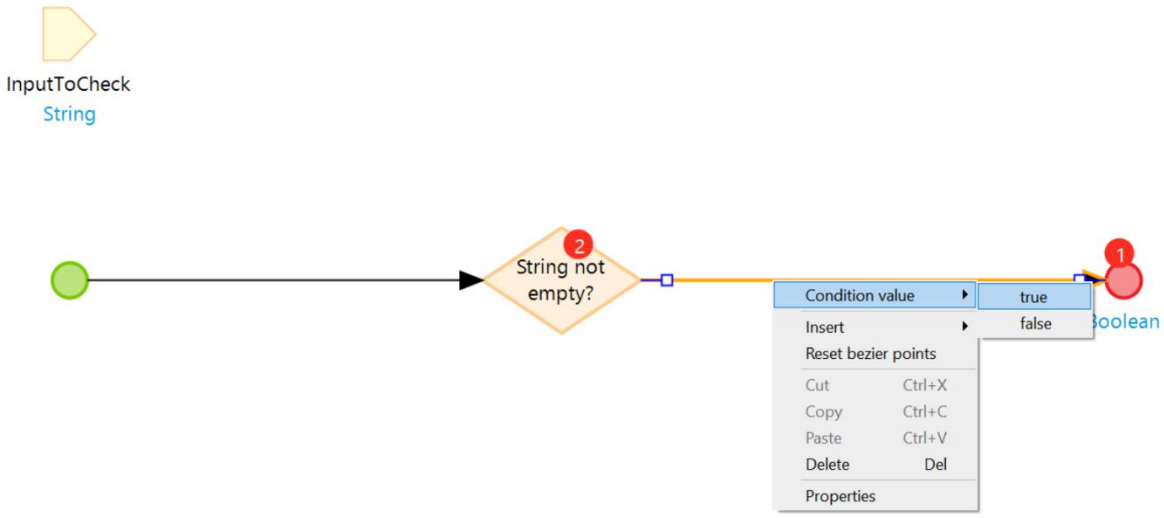


为了快捷检查许多字段是否输入有效，您可以选择使用规则。如此即可一次性编辑逻辑，并在**决策**中多次重用。规则看起来与微流非常类似，但它们有一个更受限的可用活动库，因为它们是从**决策**活动中接收输入。

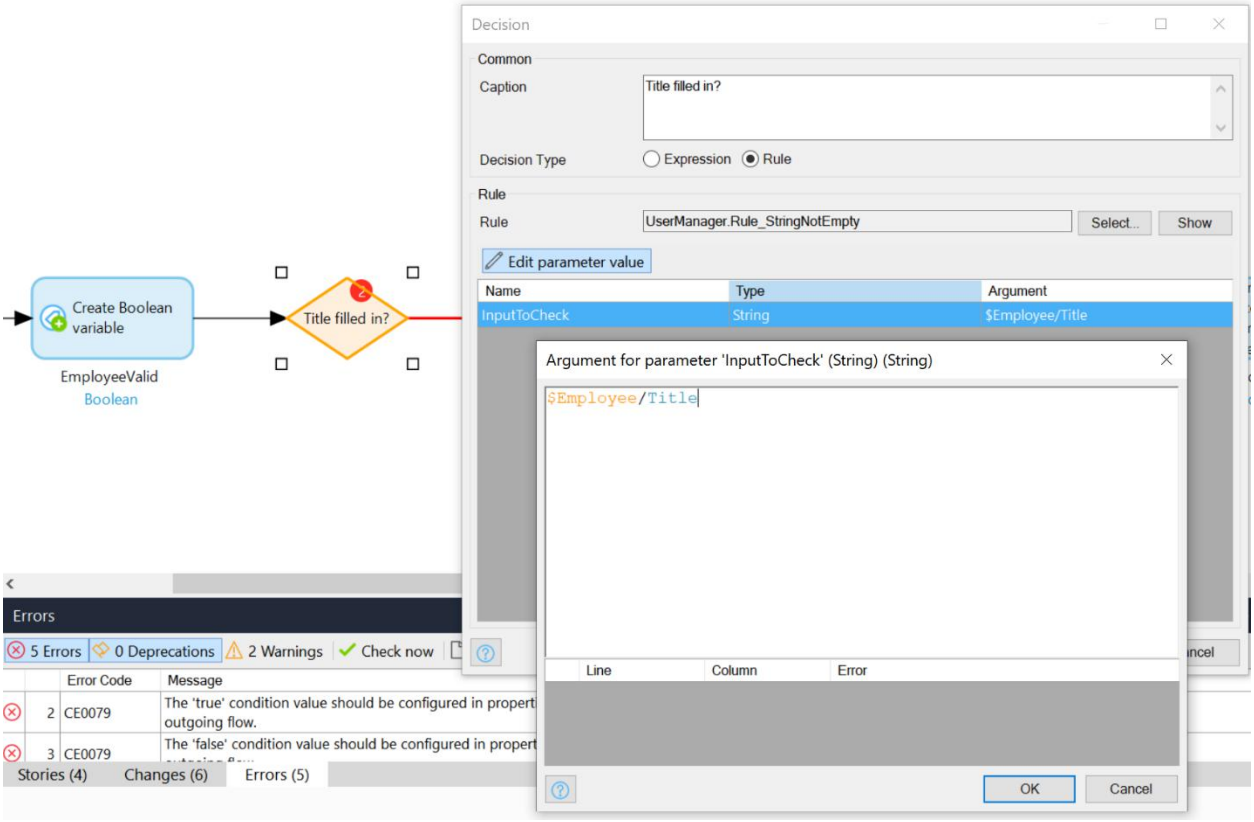
2. 右键单击 **OverviewPages** 文件夹，选择**添加其他... → 规则**，将规则命名为 **Rule_StringNotEmpty**。
3. 在新建规则中，添加参数，将**数据类型**设置为**字符串**，并将参数命名为 **InputToCheck**。
4. 在此规则中添加一个**决策**活动。右键单击此决策，将以下表达式添加到表达式窗口。将**标题**设置为**不为空的字符串？**



5. 右键单击红线，将**条件值**设置为 **true**。

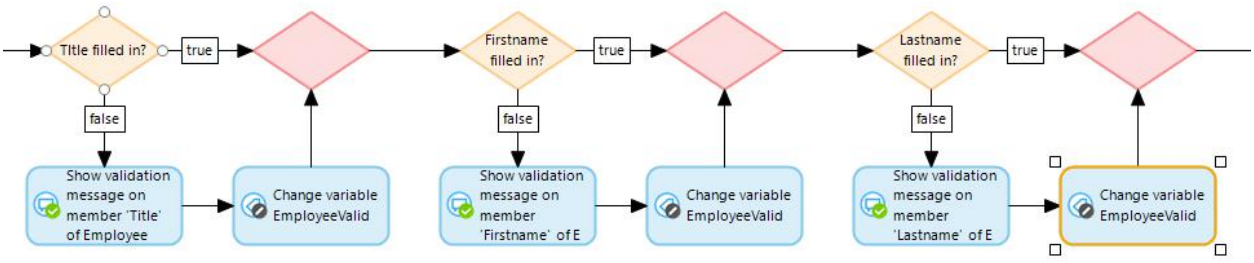


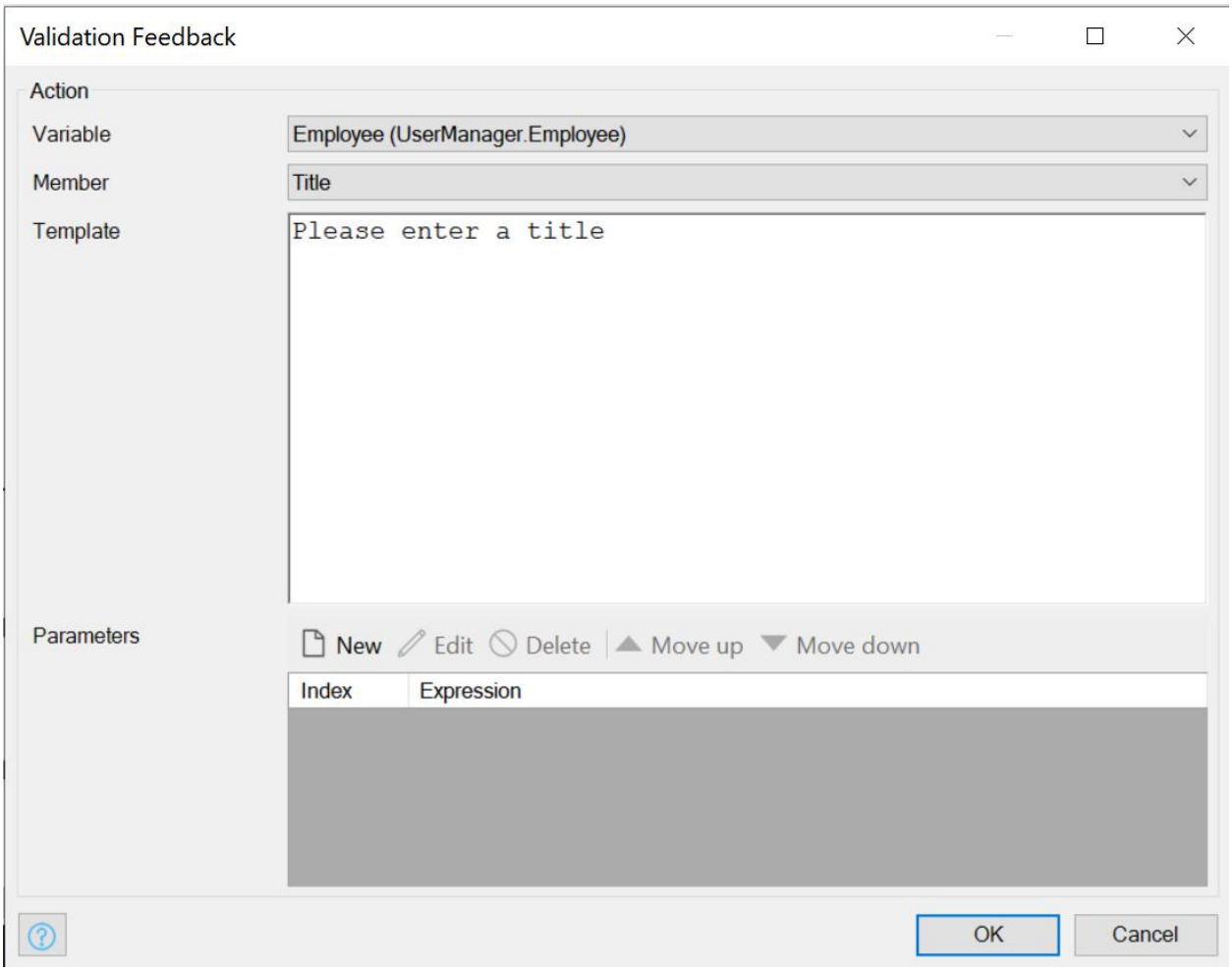
6. 双击终点，将表达式设置为 **true**。
7. 将鼠标悬停在白色圆圈上，直到它显示灰色，然后单击并向下拖动，即可从您的**决策**底部添加一个流程。找到需要流程结束的点后，释放鼠标按钮并选择**终点**作为活动。
8. 回到 ACT_Employee_Save 微流中，在**创建变量**和**提交员工**活动之间添加三个**决策**活动。
9. 对于每个**决策**，在**决策类型**中选择**规则**，选择规则 **Rule_StringNotEmpty**，然后分别将参数设置为 **$Employee/Title**、**$Employee/Firstname** 和 **$Employee/Lastname**。为每个参数设定描述性标题，然后将红色流程的**条件值**设置为 **true**。



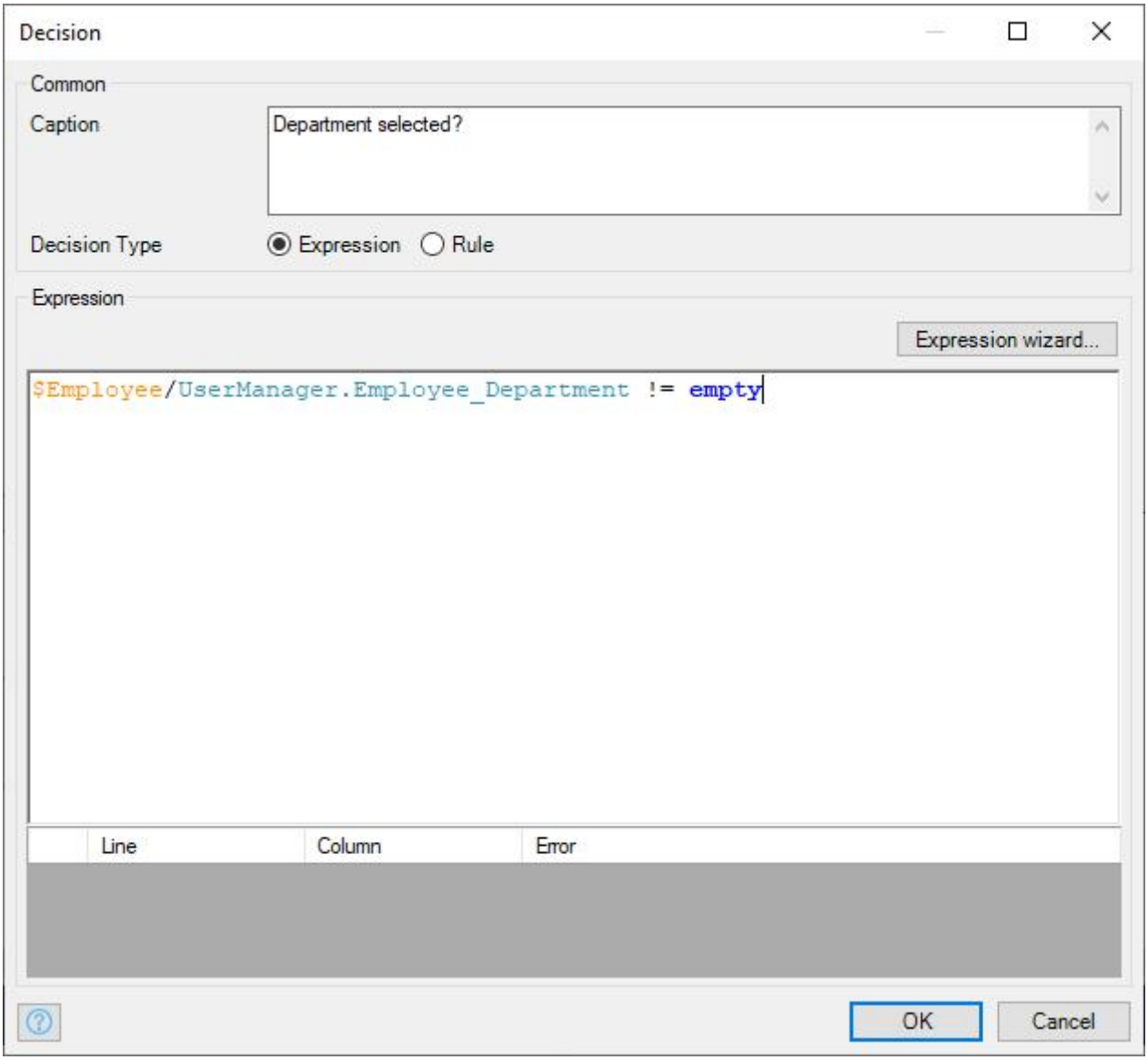
为了确保用户发现表单中的任何问题，可使用**验证反馈**活动。此活动允许您选择实体上的属性，以便 Mendix 确定在输入表单上显示验证消息的地方。

10. 将验证消息添加到您的决策的 **false** 流，并相应设置**变量**和**成员**属性。然后，添加一个**更改变量**活动，将 **EmployeeValid** 变量设置为 **false**。最后，在**决策**活动之后添加**合并**活动，以便微流继续处理。

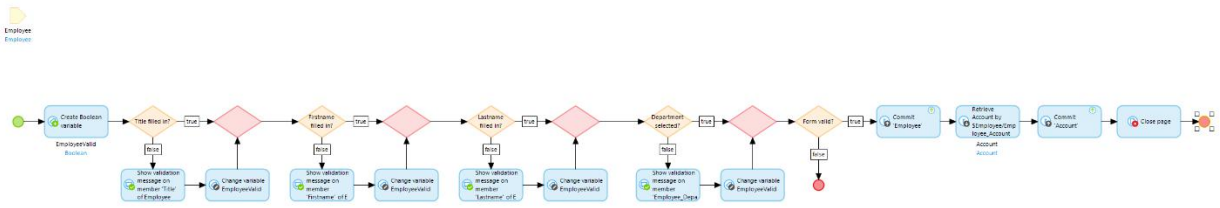




11. 添加第四个检查，确保已选择一个部门。

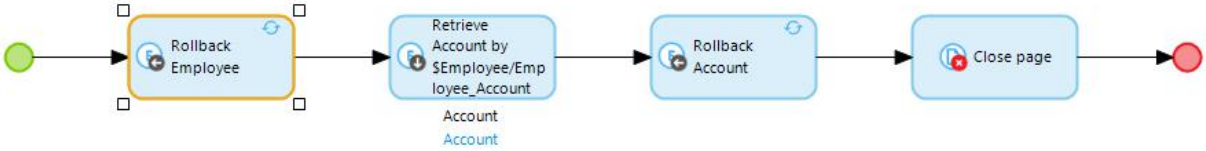


12. 在**提交员工**活动之前，添加一个最终决策，检查 **EmployeeValid** 是否成立。如果是，应提交更改，页面关闭。如果否，此流应在**结束事件**中结束。整个微流如下所示。



13. 您的 **ACT_Employee_Cancel** 微流具有相同的结构，但您想要的不是**提交**活动，而是**回滚对象**活动。那么请打开微流，继续制作。特别提醒，请您务必为两个**回滚**

对象活动勾选客户端刷新的复选框。我们正在重置实体，有必要通知客户我们的这一操作。



这就是您的前三个微流的情况。祝贺您！通过您现已创建的逻辑，用户可从 REST API 检索数据，并正确处理此应用程序中员工和帐户实体的编辑操作。

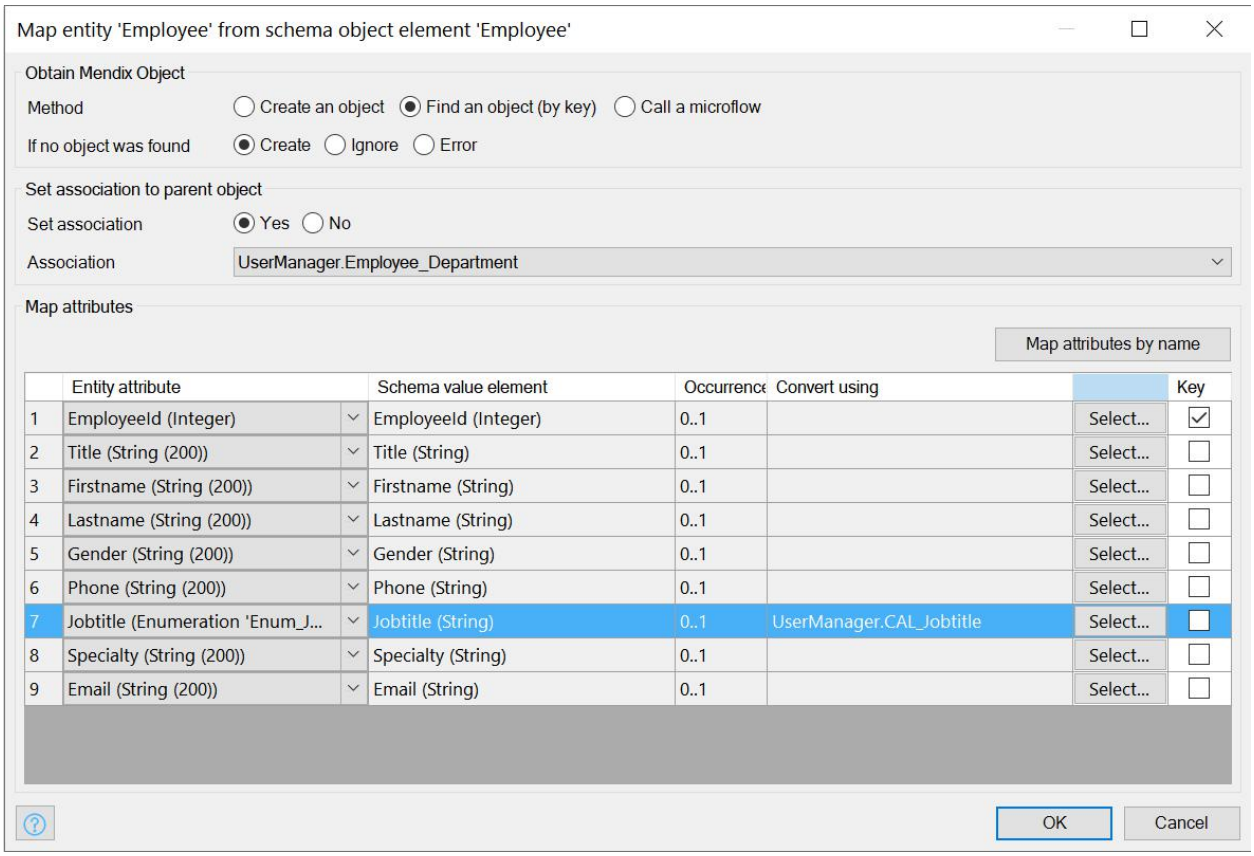
讲座 7.5：枚举

枚举是以一对字符串的形式实现的，其中一个是用于显示的**标题**，另一个是用于系统内比较的**值**。一旦创建完成**值**后，就无法再更改它。但是，标题始终可以更新。

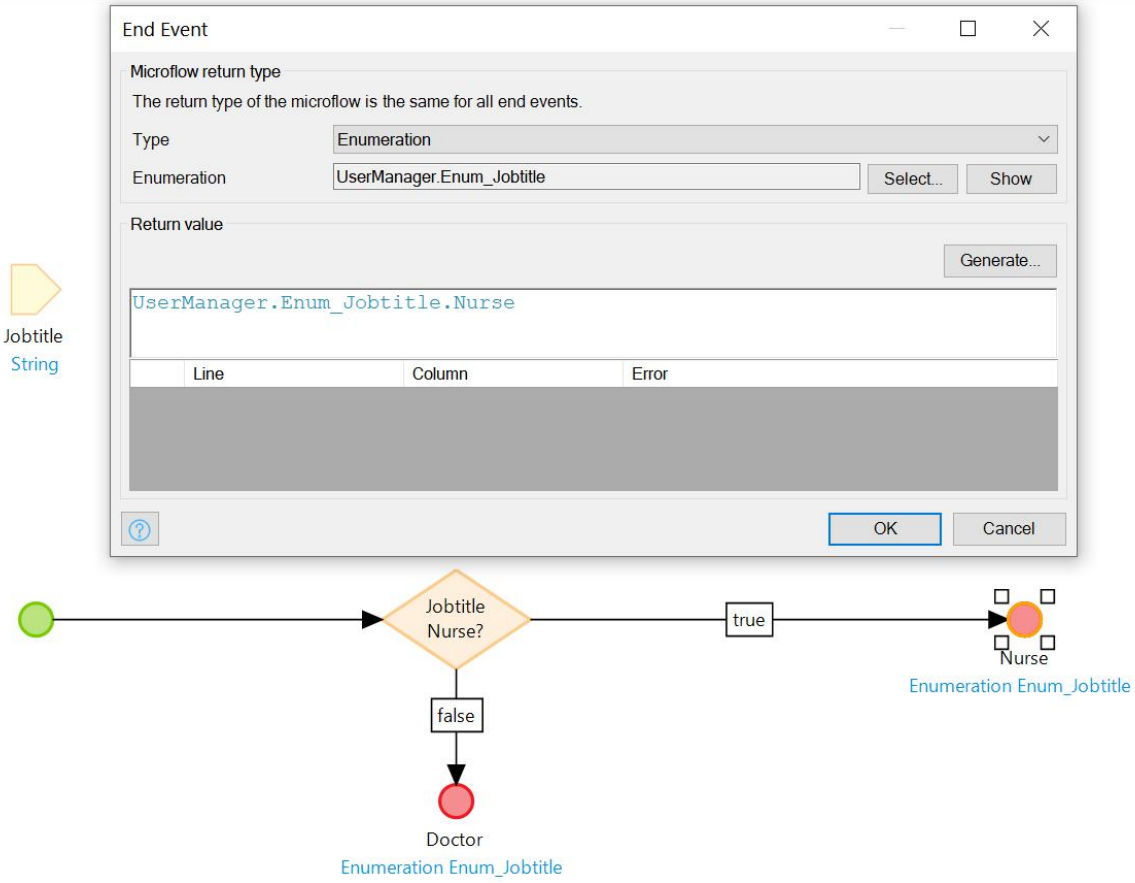
讲座 7.5.1：转换导入变量

还记得如何将字符串属性的长度更改为 200 吗？还有一些属性也可配置为**枚举**。REST 服务不提供这些枚举，因此需要转换输入。幸运的是，Mendix 在**导入**文档中提供了一个简单的方法。为此，您应首先将这些属性从**字符串**更改为**枚举**。我们现在就开始吧！

1. 从 **UserManager** 模块中打开**域模型**，然后打开**员工**的**属性**。
2. 将 **Jobtitle** 属性类型更改为**枚举**。
3. 创建一个新的枚举 **Enum_Jobtitle** 并添加**护士**和**医生**的值。
4. 打开 **ImM_Department_Employee** 实体，然后双击**员工**实体以打开属性。
5. 单击 **Jobtitle** 属性的**选择**，创建一个新的微流，命名为 **CAL_Jobtitle**。

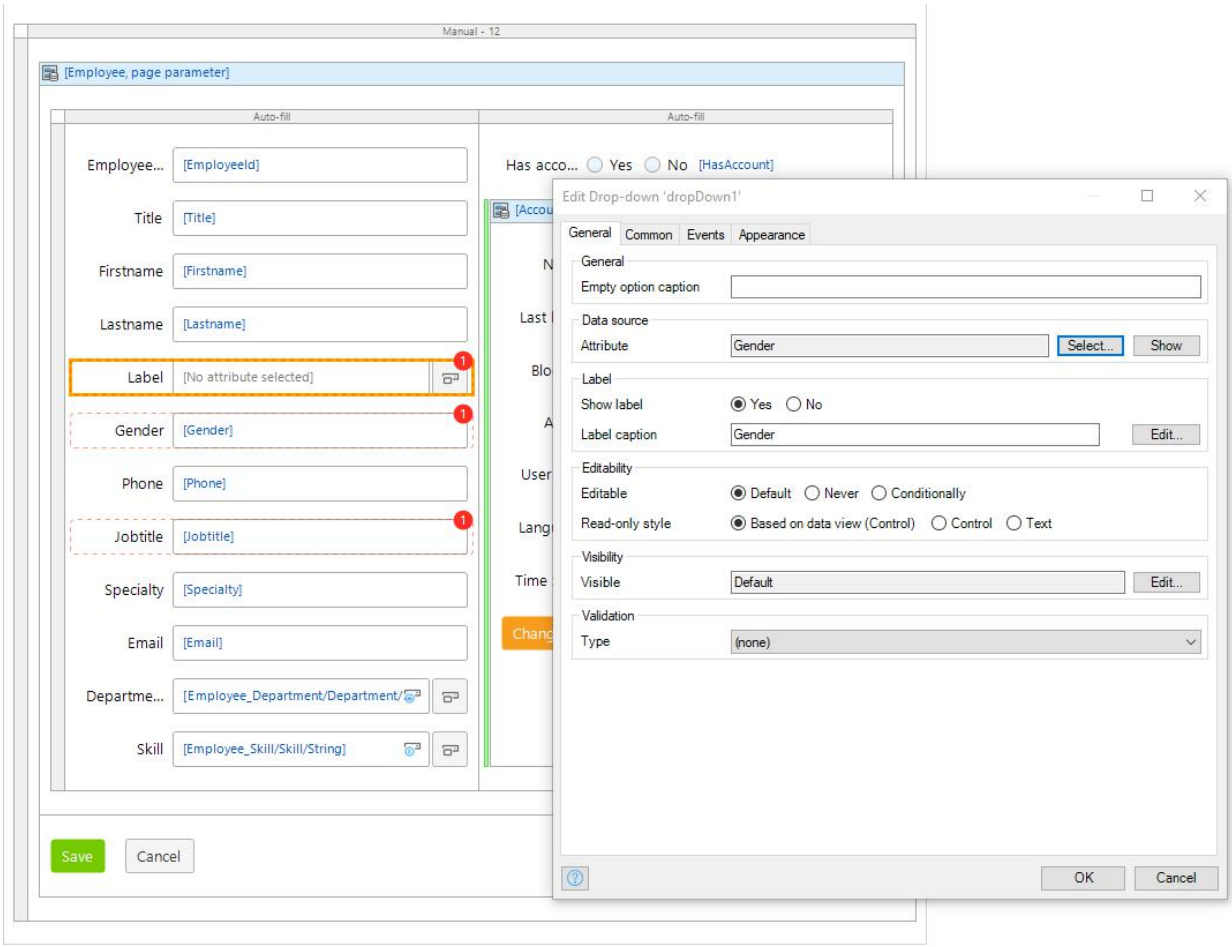


6. 在微流中添加一个**决策**，并检查 **$Jobtitle** 是否有‘**Nurse**’这个值。将**端点**的值设置为 **UserManager.Enum_Jobtitle.Nurse**。在**决策**的底部添加一个 **false** 流程，并将**终点**的值设置为 **UserManager.Enum_Jobtitle.Doctor**。

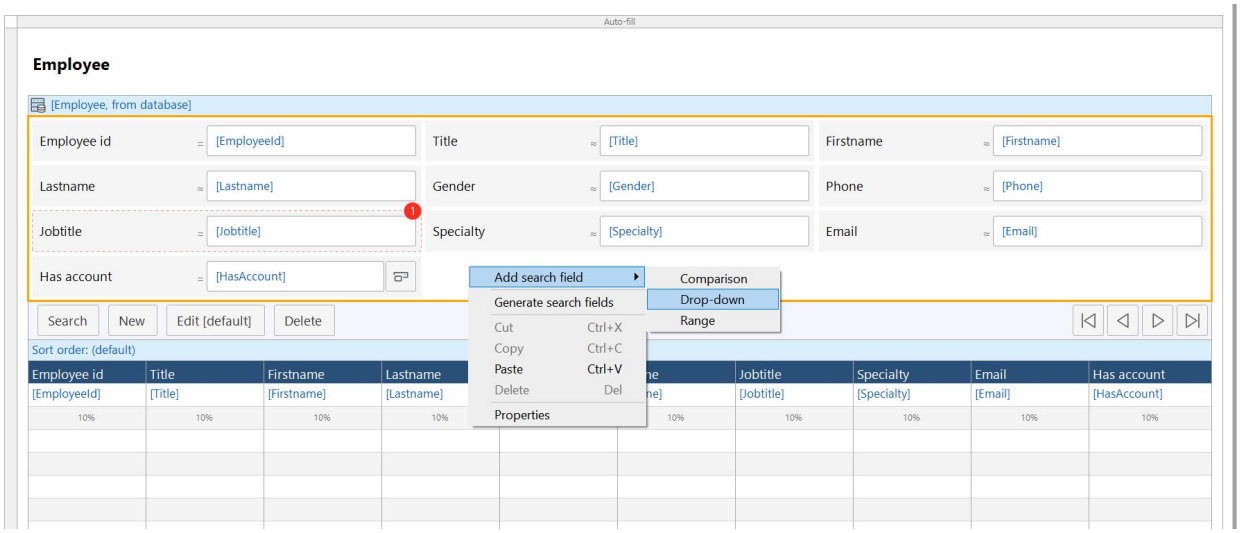


您已经更改了属性的数据类型，现在即可更新页面了。用于 **Jobtitle** 的小组件不会再这样做，所以您需要改变它。

7. 打开 **Employee_NewEdit**，将 **Jobtitle** 的输入小组件改为 **Dropdown** 小组件。选择要显示的适当属性。请记得要删除不再使用的任何小组件。



8. 移除 **Employee_Overview** 页面上的 **Jobtitle** 搜索字段，并通过右键单击将其替换为 **Jobtitle** 的 **Drop-down**。这应该能解决您之前遇到的问题。

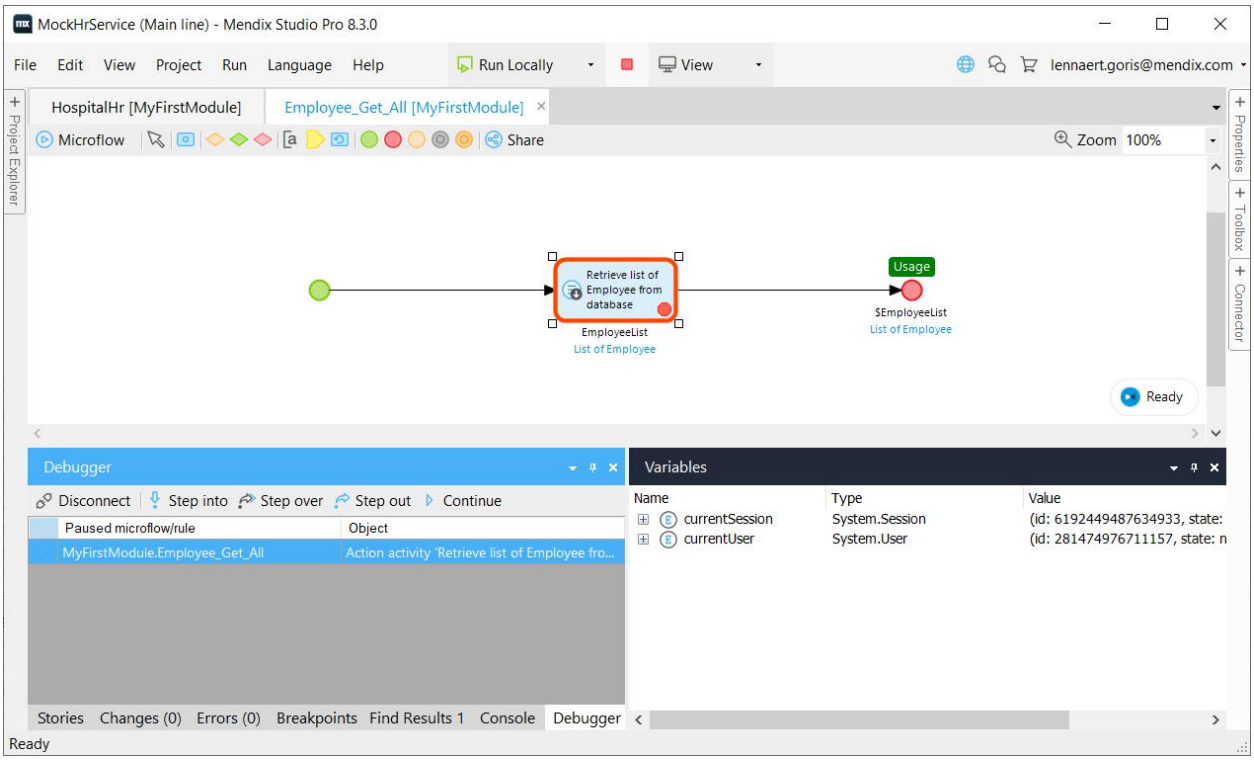


祝贺您，您又为 Summerhill 医院的人们完成了一项任务！这将帮助他们更轻松地搜索数据。

讲座 7.6：调试微流

微流编辑器自带调试程序，可供您设置断点和检查变量。它提供常见的功能，例如 **Step into**、**Step over**、**Step out** 和 **Continue**。变量视图显示范围内的所有对象。没有参数的微流有三个对象：

- currentDeviceType
- 此对象包含用于调用微流的设备
- currentSession
- 此对象包含当前会话的信息
- currentUser
- 此对象包含有关当前用户的信息



除了本地调试之外，您还可以进行远程调试。区别在于，执行远程调试时，附加至部署在云中的运行时。为此需要一个调试 URL、密码和满足调试应用程序所需的 Mendix Studio Pro 版本。另一大关键点在于，一旦执行远程调试，断点将停用。必须在 Mendix Studio

Pro 在断点停止之前手动激活。这是一项安全功能，因此请您不要停止应用程序，否则如果调试的远程应用程序正在生产中，可能会造成诸多不便。

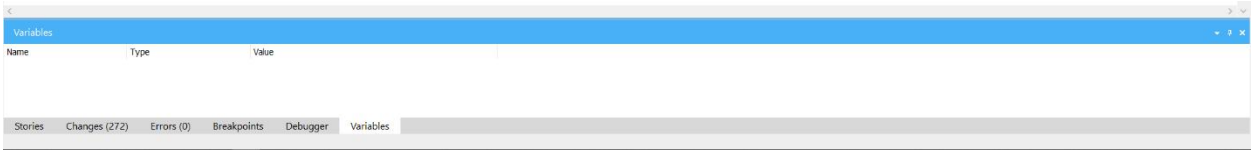
讲座 7.6.1：设置调试程序

您已经了解到微流调试程序的功能，现在是时候使用它了。接下来我们来看看执行微流时它们会发生什么，探索调试程序中的哪些选项可以为您所用。现在，您将在本地运行调试程序，但请记住，您还可以将调试程序附加到在云中运行的应用程序。让我们设置您的调试程序。

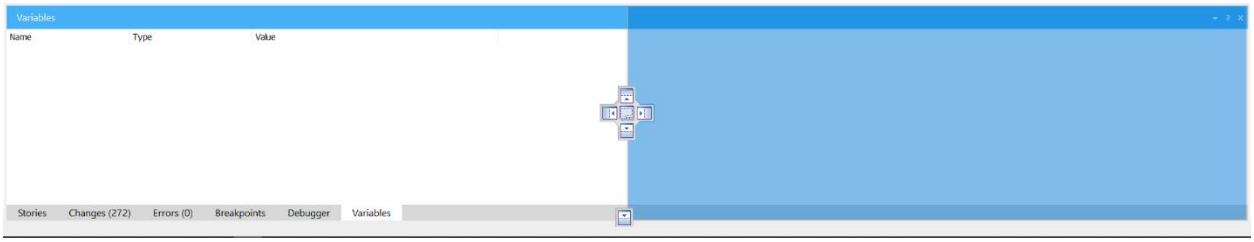
1. 打开 **ACT_Department_Employee_ImportFromRest** 微流。
2. 单击**查看 > 调试窗口 > 断点**
3. 单击**查看 > 调试窗口 > 调试程序**
4. 单击**查看 > 调试窗口 > 变量**

现在您已经激活了所有需要的窗口，让我们看看它们的位置并进行设置，以便轻松查看我们正在运行的微流的所有不同方面。

5. 定位屏幕底部的视图，向上拖动使窗口变大。



6. 拖动选项卡并放在所显示的十字形选择器的右侧，将**变量**窗口向右移动，如屏幕截图所示。



7. 选择**调试程序**选项卡。

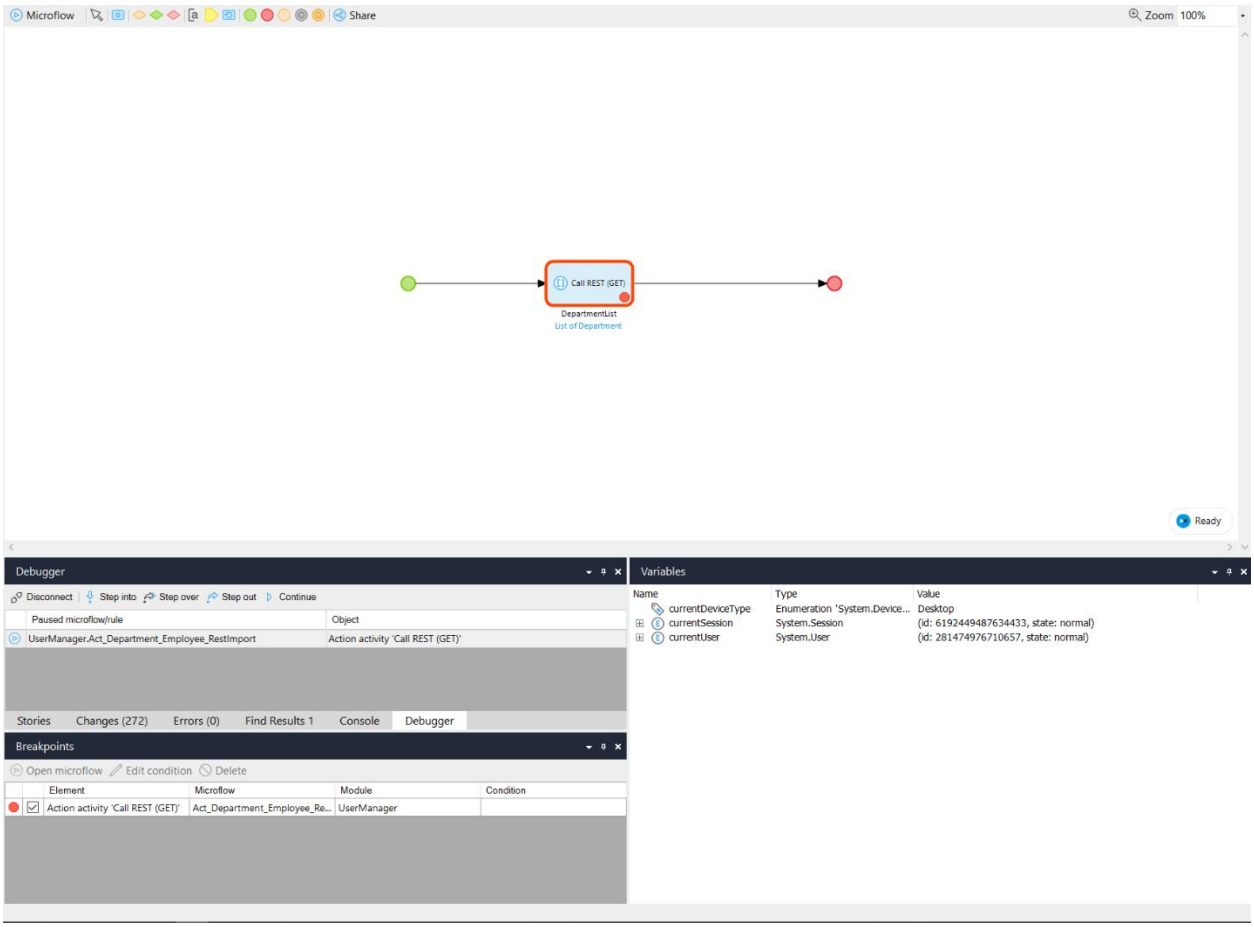
很棒，您已完成调试程序的设置，可以看到所有相关的调试屏幕。若出于某种原因想要重置窗口，可通过**查看 → 重置布局…**来实现您也了解到如何使用 Mendix Studio Pro 中的对接功能，可重新配置用户界面以满足您的需求。

讲座 7.6.2：调试微流

现在调试屏幕已设置完成，可运行调试程序，在微流执行过程中查看它的各个方面。

1. 右键单击您的导入活动，选择**添加断点**。
2. 确保 **MockHrService** 正在运行。
3. 在 Summerhill 项目中单击**本地运行**。
4. 项目运行后，单击**查看**。
5. 在运行 Summerhill 应用程序的过程中单击您的导入按钮。

您的微流将在导入活动中停止。您会看到，计算机任务栏上的 Mendix Studio Pro 图标改变了颜色。若现在切换到 Studio Pro，会显示以下屏幕。现在，您可以通过使用**变量**窗口和**调试程序**窗口中的各种**步骤**命令来检查微流中每一步在当前范围内的变量内容。您可以右键单击现有的断点来移除或禁用它们。



祝贺您，您已经成功调试您的第一个微流，并探索了 Mendix 又一个伟大的功能！让我们继续下一个主题。

讲座 7.7：无状态运行时

运行时本身仅包含操作持续时间内的状态。操作完成后，无论是保存等默认操作还是微流等定制操作，运行时均会向数据库发送所有已提交对象。所有非永久对象、所有未提交的

对象和所有已提交的对象，客户端在下一个页面所需要的全部对象，均会发送到客户端。这使得运行时保持无状态。因此，Mendix 有能力进行横向扩展。如果您有多个运行时在负载均衡器后面运行，那么哪个运行时被客户端击中并不重要；它们都是平等的，因为它们都是无状态。这种无状态也有助于轻松创建和销毁运行时实例 — 如果需按照十二要素原则构建一个应用程序，这是个关键要求。

这也确实引出了一个问题：非永久对象的生命周期是怎样的？它只在执行操作时存在于运行时，其余时间都在客户端，那么客户端决定了非永久对象的生命周期。它们将通过在客户端上创建并启动生命周期。如需在服务器上执行，它们会被发送到运行时进行处理，但总是会送回。鉴于这一事实，非永久对象唯一能够生存的位置是客户端。如果非永久对象被手动删除或垃圾回收，原因在于没有更多的对象引用（指 Mendix 关联），或客户端被关闭，非永久对象将被销毁。

总结

在这个模块中，您执行了很多操作！您了解到什么是微流、什么是微流元素，并且已经见识了一些基本的微流活动。您已设置好 Mendix Studio Pro 以轻松调试应用程序，并已在调试程序中执行了您的第一个微流。您现已掌握了许多内容，全方位涵盖基础知识。您已准备好构建应用程序！但在部署之前，务必要确保数据安全，因此我们接下来谈一谈安全问题。