outsystems

Form Validations - How-to

Table of Contents

Outline

In this exercise, we will focus on adding some business logic to our application, by validating the forms we have already created. This will be done in the MovieDetail and PersonDetail Screens to avoid saving invalid information in the database.

At the end of this exercise, we want to guarantee that the following rules are followed:

- A movie's gross takings can never be a negative number.
- A movie cannot have any gross takings amount, if it wasn't released yet.
- A person's birth date cannot be in the future.
- A person's date of birth cannot be later than the person's date of death.

If a movie or person was successfully added to the database, then a feedback message should appear to the user.

How to

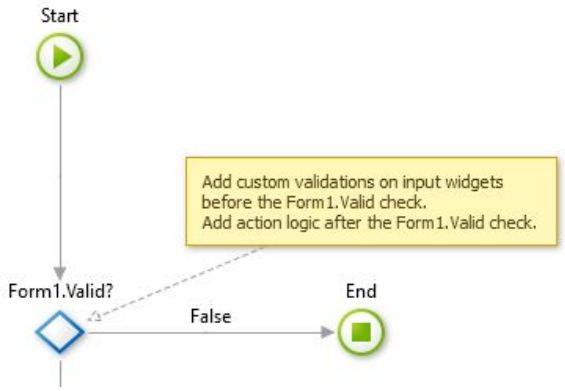
In this section, we'll describe, step by step, the exercise *5.3 - Form Validations*.

MovieDetail Screen Validations

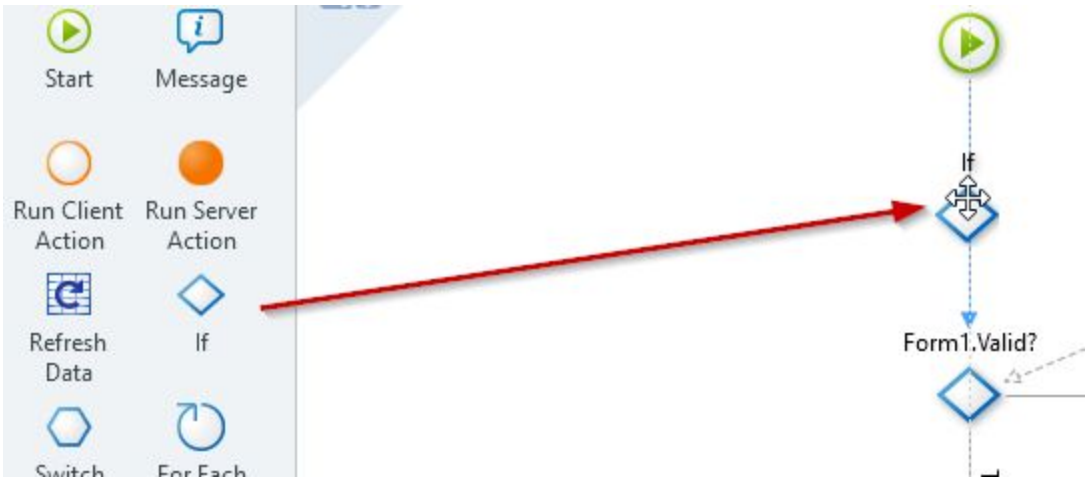
We will start this how-to by implementing the first two validations, which refer to the MovieDetail Screen.

1. In the SaveOnClick Action of the MovieDetail Screen, create the logic to validate that a movie gross takings can never be a negative number.

    a. Open the **SaveOnClick** Client Action from the MovieDetail Screen**.**



b. Drag an **If** to the Action flow and drop it before the existing If.

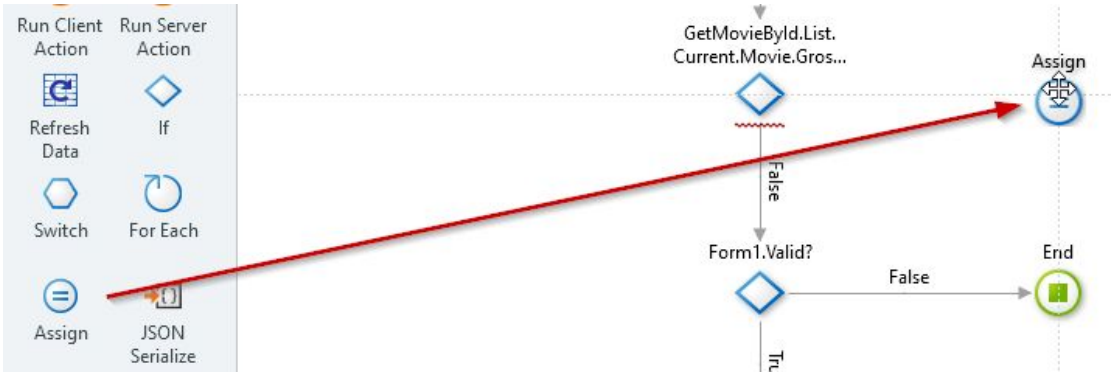


c. Set the **Condition** of the If to be:

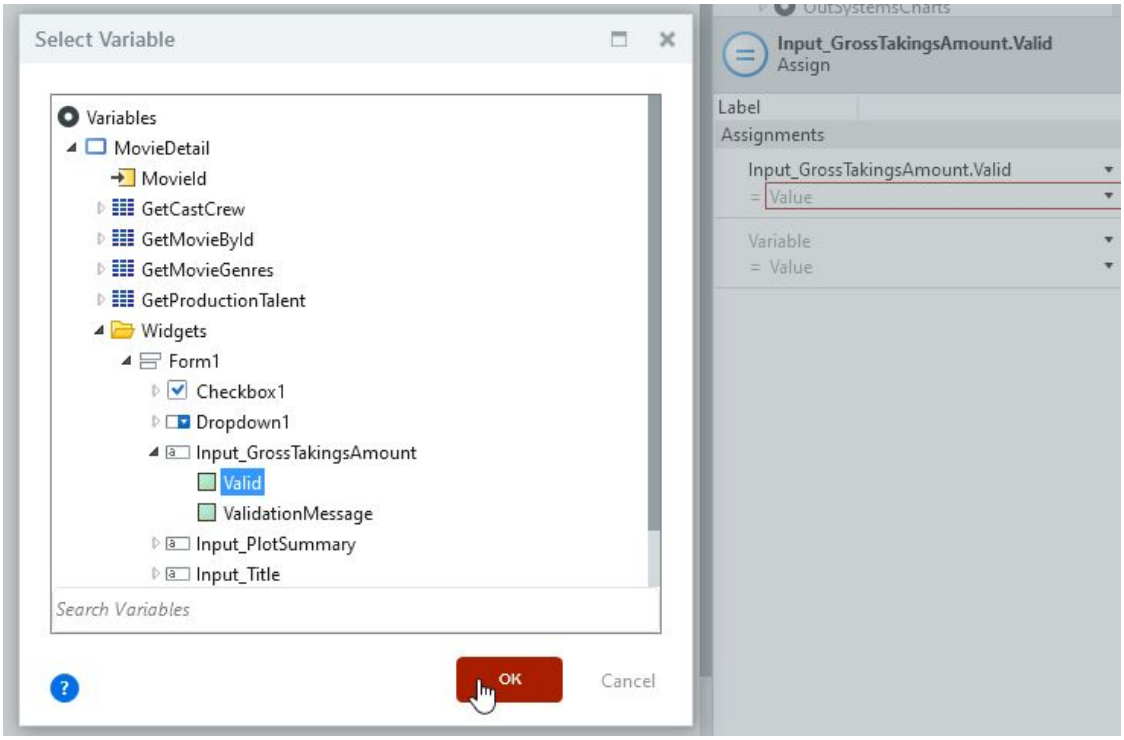
GetMovieById.List.Current.Movie.GrossTakingsAmount < 0

Currently the If still has an error, since the True connector is missing. We will fix it later.

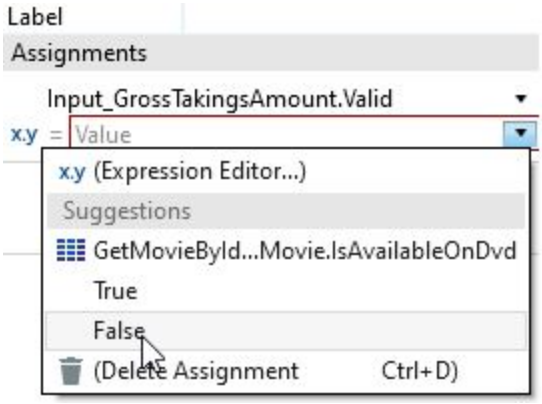
d. Drag an **Assign** and drop it next to the recently added If.



e. In the Assign node, select the first dropdown. In the new dialog, choose the **Valid** property of the GrossTakingsAmount input field, which can be found under the WIdgets folder.



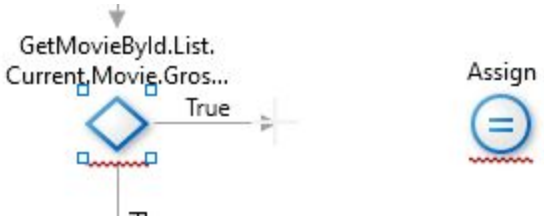
f. Set the **Value** for the Assign to be *False*.



g. In the same Assign node, create the following assignment after the previous one

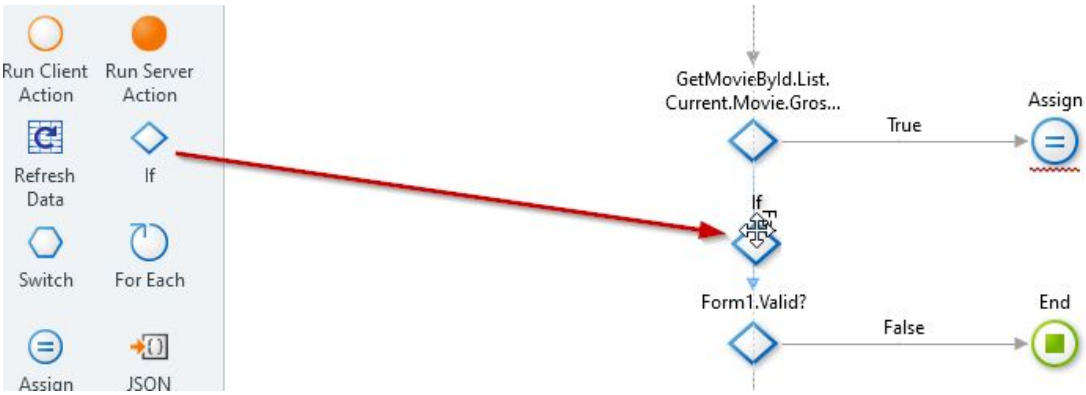
Input_GrossTakingsAmount.ValidationMessage = "Gross Takings Amount cannot be negative."

h. Connect the If to the Assign node. This creates the True branch, indicating that if the Condition is true, it means the data is not valid.



2. In the same Action, create the logic to validate that a movie cannot have any gross takings if it wasn't released yet.

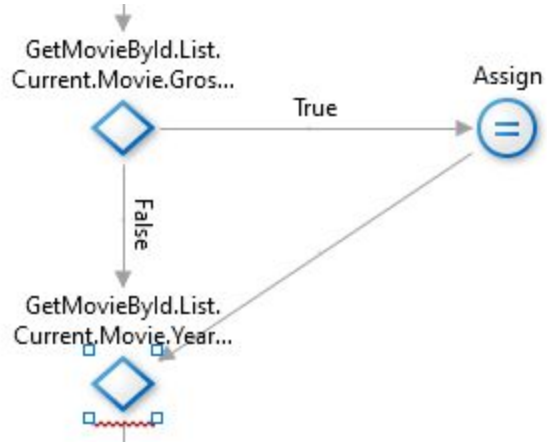
   a. Drag another If between the two existing ones.



   b. Set its **Condition** to

      GetMovieById.List.Current.Movie.Year > Year(CurrDate()) and GetMovieById.List.Current.Movie.GrossTakingsAmount > 0

   c. Connect the Assign to the new If



d. Drag a new Assign and drop it to the right of the recently added If. Connect the two nodes together and define the following assignments

Input_GrossTakingsAmount.Valid = False
Input_GrossTakingsAmount.ValidationMessage = "The movie cannot have any gross takings since it was not yet released."

e. Connect the new Assign to the Form1.Valid? If.

f. Publish the module and test the validations in the browser.

PersonDetail Screen Validations

Now, we will move to the PersonDetail Screen and implement the two missing validations.

1. In the SaveOnClick Action of the PersonDetail Screen, create the validation logic to guarantee that the person's birth date cannot be in the future.

   a. Open the **SaveOnClick** Action of the PersonDetail Screen.

   b. Drag an **If** to the Action flow and drop it below the Start node. Set its **Condition** to

   GetPersonById.List.Current.Person.DateOfBirth > CurrDate()

   c. Drag an **Assign** node next to the If and set the following assignments. Don't forget to connect the If to the Assign.

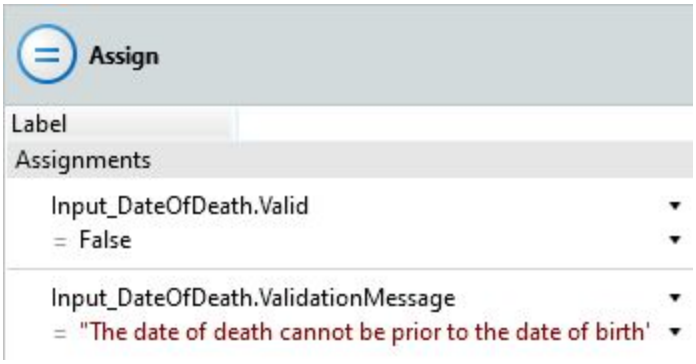
2. Now, create the validation logic to guarantee that the person's date of death cannot be before the date of birth.

a. Drag another **If** between the two existing Ifs and set the **Condition** to

   GetPersonById.List.Current.Person.DateOfDeath <> NullDate() and GetPersonById.List.Current.Person.DateOfDeath < GetPersonById.List.Current.Person.DateOfBirth

b. Connect the Assign for the Date Of Birth in the future with the recently created If

c. Drag a new **Assign** node next to the If and set the following assignments. Don't forget to connect the new If with this new Assign



d. Connect the Assign with the **Form1.Valid?** If

e. Publish the module and test the new validations.