

Security and Exceptions - How-to

Table of Contents

Outline	2
How to	2
Create Users and Roles	2
Check Roles in Logic	4
Restrict Access on Screens	12
Disable Inputs in Screens	15

Outline

In this exercise, we will add some security features to the OSMDb application. So far, all the Screens can be accessed and operations can be completed with the Anonymous role, meaning that there are no security restrictions. At the end of this exercise, we want to make sure that only Admins can create/update information for movies and people, including adding a cast or crew member to a movie.

We want to do this in several steps:

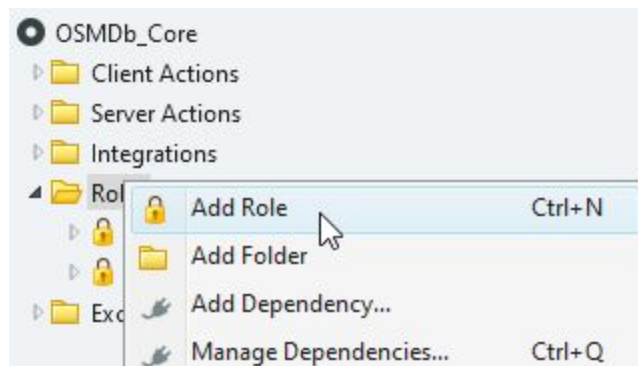
1. Create a OSMDbAdmin role.
2. Create some end-users and grant the role to one of them.
3. Guarantee in the logic that only OSMDbAdmin users can add a member of cast/crew to a movie. Use Exceptions to display an error message to the user if they do not have the correct permissions.
4. Filter the access to the AddCastAndCrew Screen to OSMDbAdmin users.
5. Make sure that only Admin users can edit the movie and people's information.

How to

In this section, we'll describe, step by step, the exercise 6.3 - *Security and Exceptions*.

Create Users and Roles

1. Create a new Role, OSMDbAdmin, in the OSMDb_Core module.
 - a. Open the **OSMDB_Core** and the Logic tab.
 - b. Right click the Roles folder and choose **Add Role**



- c. Name it *OSMdbAdmin* and make it **Public**

OSMdbAdmin Role	
Name	OSMdbAdmin
Description	...
Public	Yes
Advanced	
Is Persistent	Yes

- d. **Publish** the module.

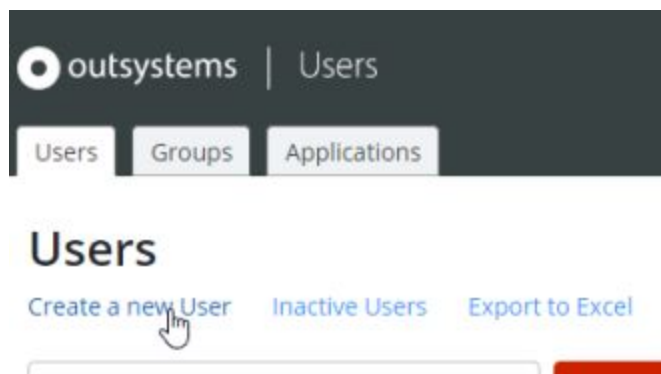
2. Go to the Users application and create two new users: one with the OSMdbAdmin Role and another one without it.

- a. Open the **Users** App in the browser by navigating to *https://<environment>/Users*
- b. Login with the following credentials:

Username: *admin*

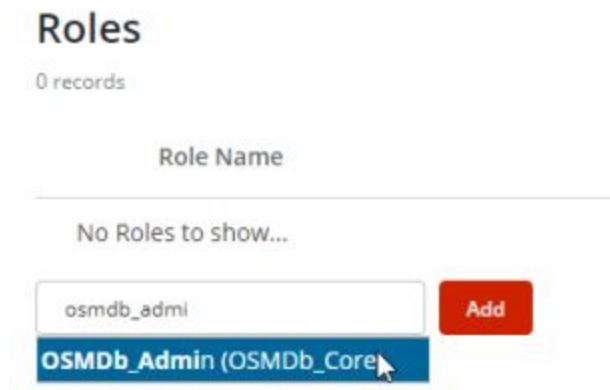
Password: *outsystems*

- c. Click on **Create a new User**.



- d. Fill the mandatory data with any info you want and set it a simple password, for example, *123456*. Click **Save**.

- e. Scroll down to the Roles area. In the dropdown, search for your **OSMDBAdmin** role (remember that it should be the one associated with your Core module) and click **Add**.

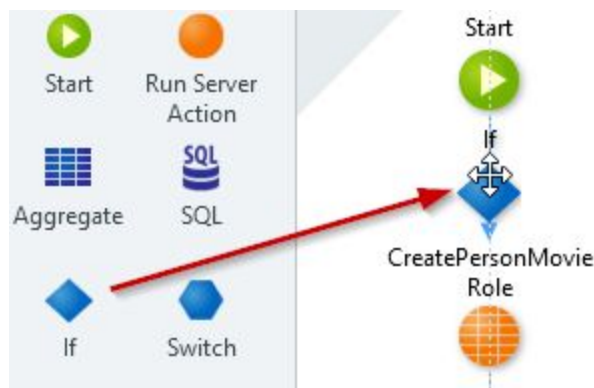


- f. Create a new user, but this time without any Role associated with it.

Check Roles in Logic

Now that the application has a new role, we will use it to expand the logic and make sure that only users with the OSMDBAdmin role can create/update records in the database.

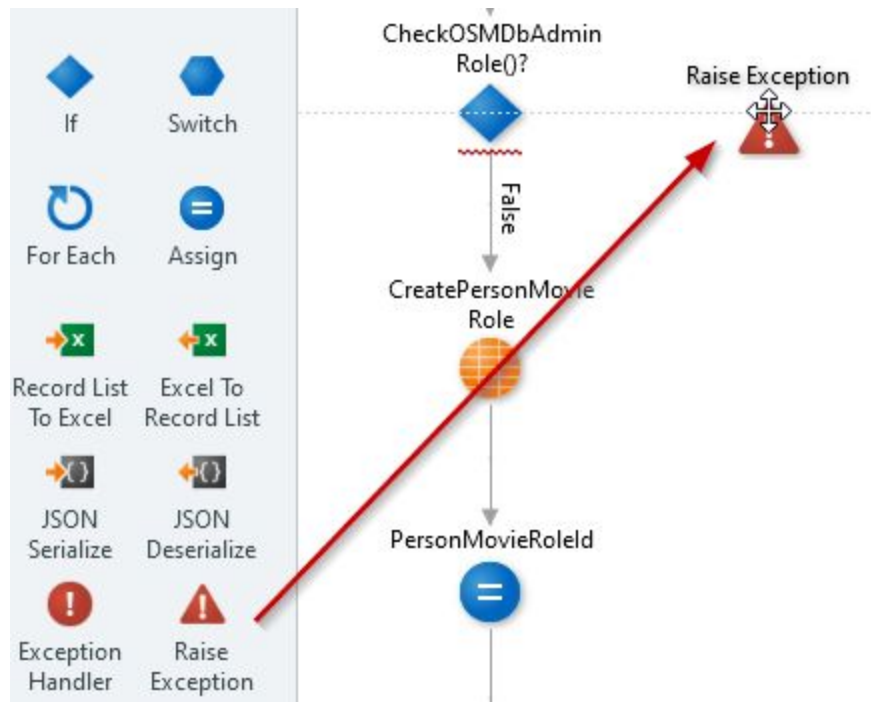
1. In the **PersonMovieRoleCreate** Action, make sure that only users with the OSMDBAdmin Role can create the record in the database. The Action should return if the operation was successful or not and a success/error message.
 - a. Open the PersonMovieRoleCreate Action, drag an **If** and drop it after the Start node.



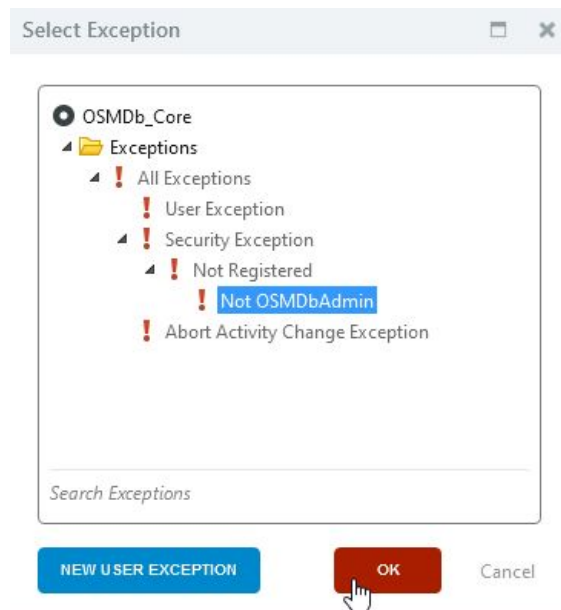
- b. Set the **Condition** of the If to *CheckOSMDBAdminRole()*

NOTE: The CheckRole Action can be used in Server Actions to verify if a certain user has that particular Role. When the CheckRole Action is called without any input parameter, the verification is done considering the user currently logged in.

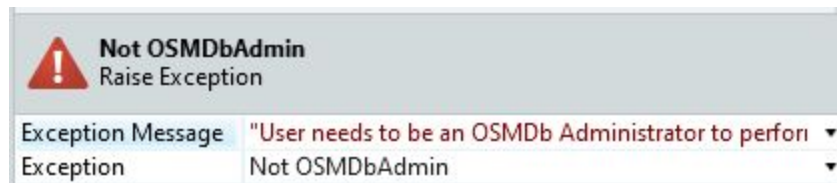
- c. Drag a **Raise Exception** statement and drop it next to the If.



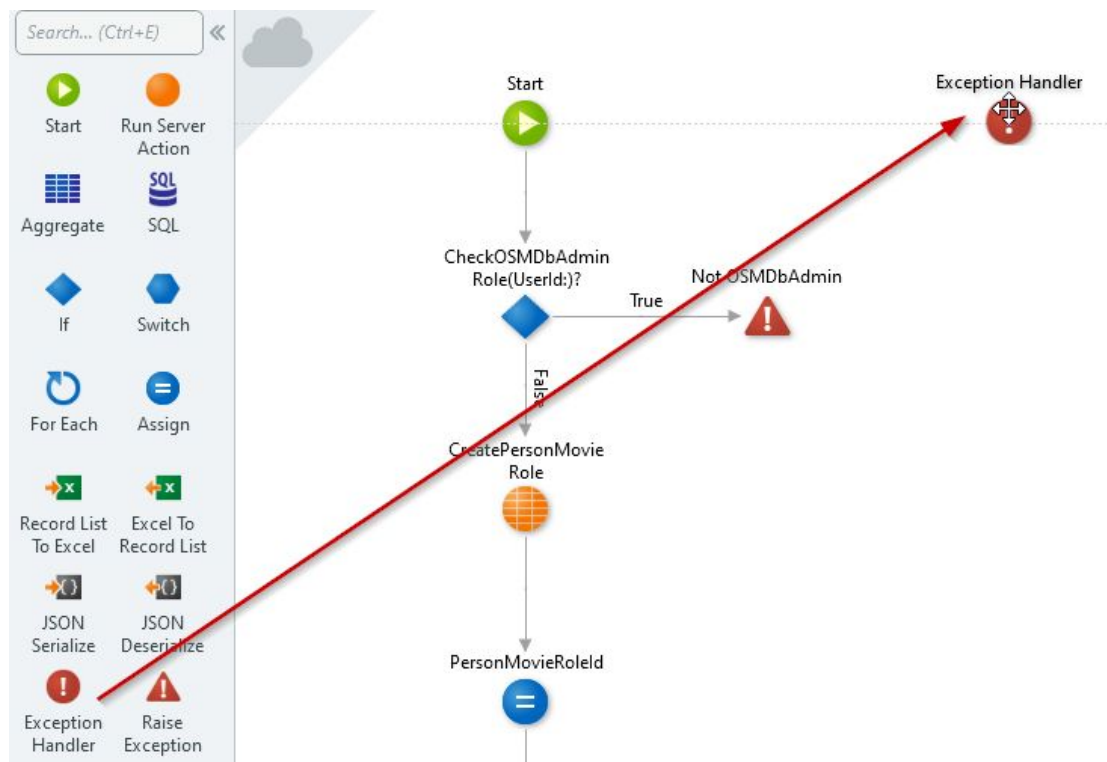
- d. In the new window, choose the **Not OSMDBAdmin** Exception.



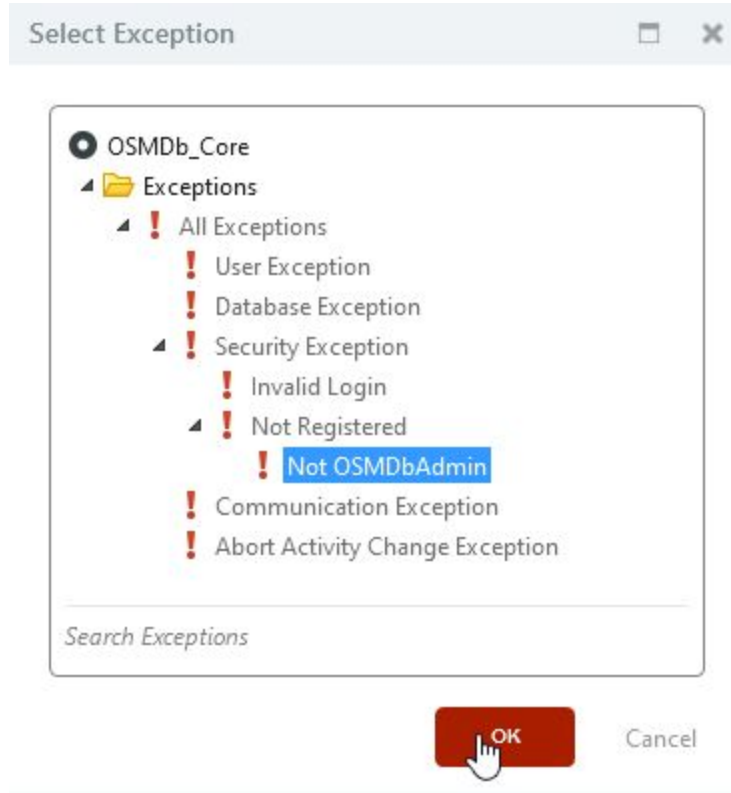
- e. Set the **Exception Message** property to "User needs to be an OSMDb Administrator to perform this operation"



- f. Connect the If to the Raise Exception.
- g. Add a new Output Parameter to the PersonMovieRoleCreate Action and call it *Success*. Make sure its **Data Type** is set to *Boolean*.
- h. Add another Output Parameter to the Action and call it *Message*
- i. Drag an Exception Handler and drop it next to the main flow of the Action.



- j. In the new window, select the **Not OSMDbAdmin** Exception.



- k. Drag an **Assign** and drop it on the Exception flow



- I. Define the following assignments

Assign	
Label	
Assignments	
Success	▼
= False	▼
Message	
xy = NotOSMDBAdmin.ExceptionMessage	▼

NOTE: When we define the Message on the Raise Exception node, that Message can then be used directly by referencing the Exception and the property name (ExceptionMessage).

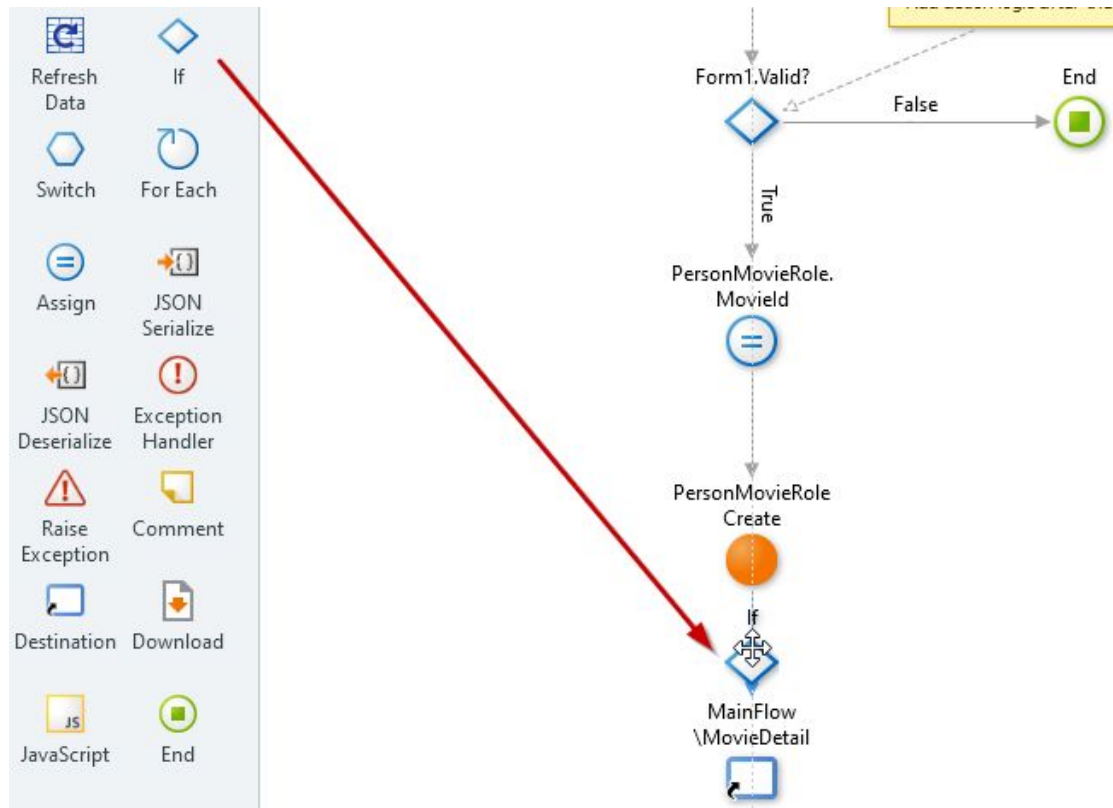
- m. Right-click on the If and select **Swap Connectors**.
- n. Select the Assign right before the End node on the main Action flow and add the following assignments

Success = True

Message = "Record created successfully"

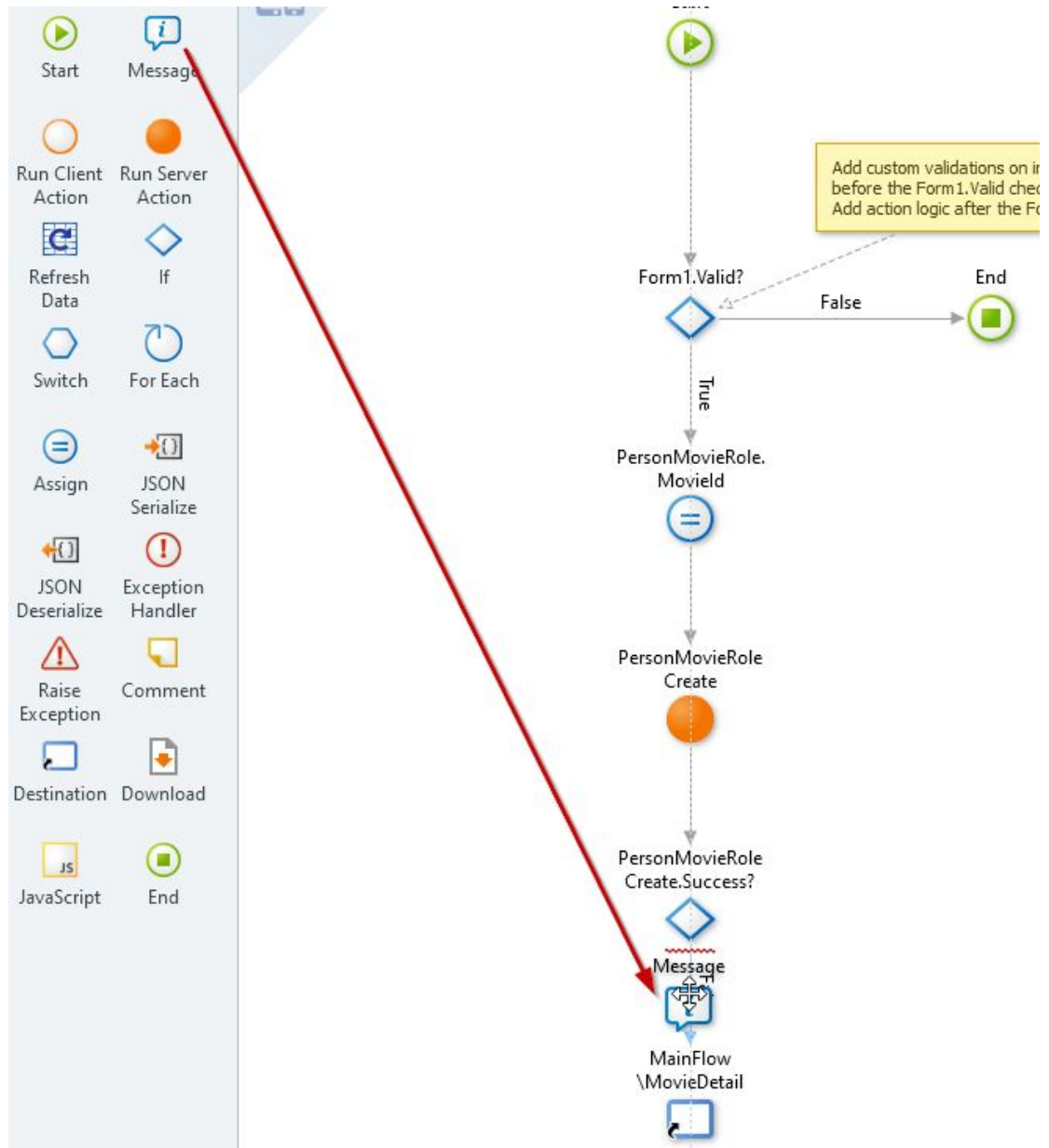
2. Repeat the exact same steps for the **MovieCreateOrUpdate** and the **PersonCreateOrUpdate** Actions.
3. Publish the module.
4. Switch back to the OSMDB module and refresh the dependencies.
5. In the SaveOnClick Action of the AddCastAndCrew Screen, use the output parameters of the PersonMovieRoleCreate Action to create some logic to display feedback messages to the end-user.
 - a. Open the **SaveOnClick** Action in the AddCastAndCrew Screen.

- b. Drag an **If** and drop it below the PersonMovieRoleCreate Action.



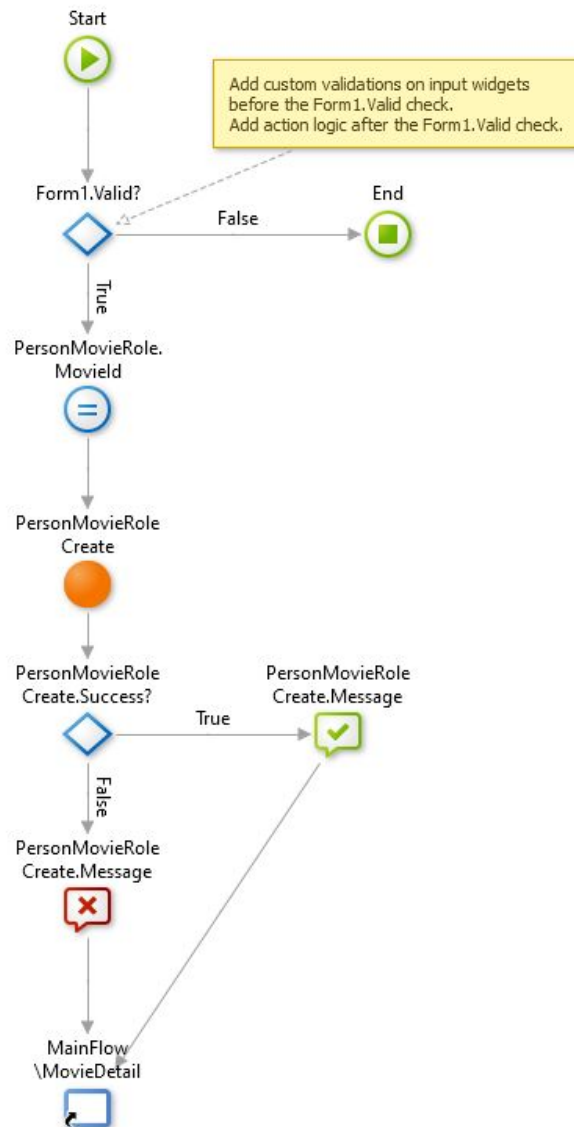
- c. Set the **Condition** of the If to *PersonMovieRoleCreate.Success*

- d. Drag a **Message** and drop it below the If. Set the **Message** property to the PersonMovieRoleCreate Message output and the **Type** to **Error**.



PersonMovieRoleCreate.Message Message	
Label	
Message	PersonMovieRoleCreate.Message ▼
Type	Error ▼

- e. Drag a **Message** and drop it to the right of the If. Connect the If to the new Message and the new Message to the MovieDetail Screen Destination.
- f. Set the **Message** property to be the output parameter of the PersonMovieRoleCreate Action and the **Type** to *Success*. The flow should look like the following screenshot.



6. Repeat the same logic above to the SaveOnClick Actions in the MovieDetail and PersonDetail Screens.

Restrict Access on Screens

In the previous section, we added some logic to check the users role in an Action flow. In this section, we will use the Roles to restrict access to Screens or to a particular feature of a Screen. Let's start with the Screens: only OSMDbAdmin users can access the AddCastAndCrew Screen. Then, the *Add Cast and Crew to Movie* Link in the MovieDetail should be hidden to all users without the OSMDbAdmin Role.

1. Select the AddCastAndCrew Screen and in the properties are, under Roles, remove the access to the **Anonymous** and **Registered** Roles.

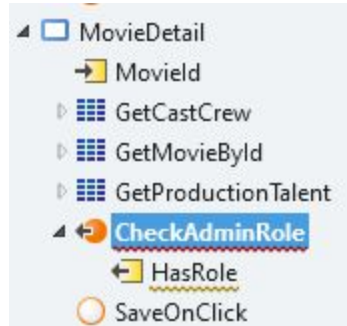
AddCastAndCrew Screen	
Name	AddCastAndCrew
Description	
Title	
Public	No
Roles	
Anonymous	<input type="checkbox"/>
Registered	<input type="checkbox"/>
OSMDbAdmin	<input checked="" type="checkbox"/>

NOTE: Make sure that the OSMDbAdmin Role is referenced from the Core module, otherwise it will not appear on the Screen properties.

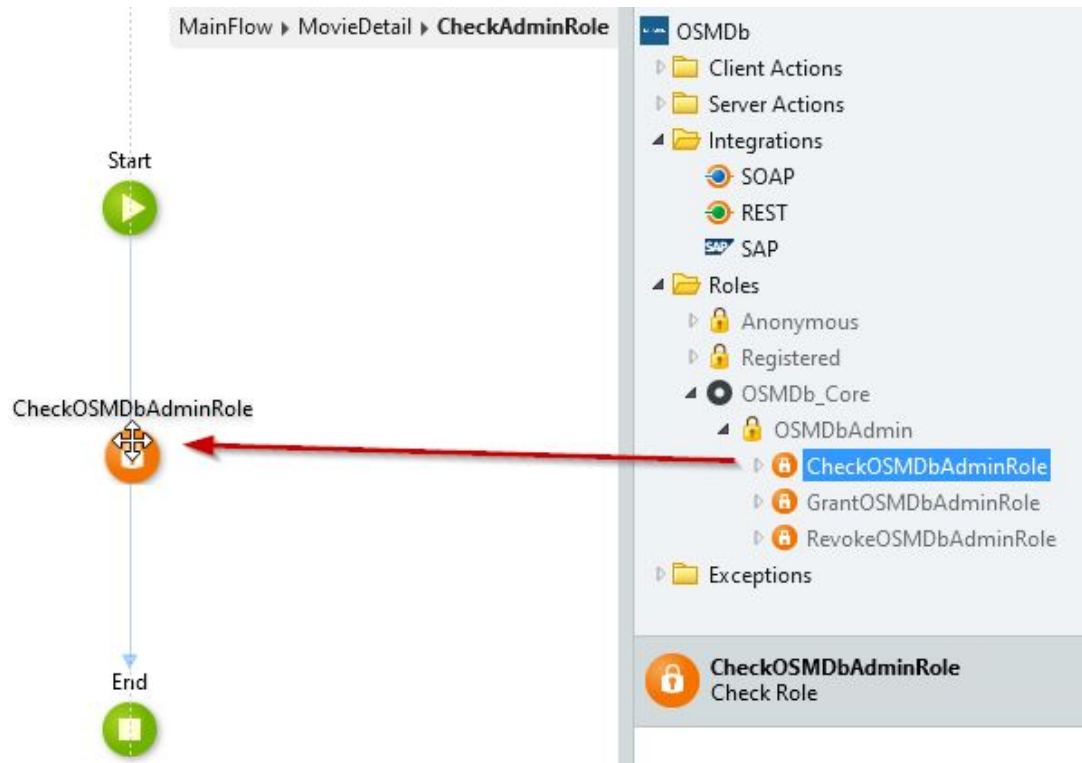
2. Now, we also need to show the **Add Cast and Crew** link in the MovieDetail Screen only to OSMDbAdmin users.
 - a. Right-click the MovieDetail Screen and select **Fetch Data from Other Sources**.

NOTE: Since the CheckRole Action is server-side, we need to use a Data Action to determine if the user has the expected role.

- b. Name it *CheckAdminRole* and change the output parameter to *HasRole*



- c. Open the Data Action and switch to the Logic tab. Under **Roles->OSMDb_Core**, you will find the OSMDbAdmin Role. Drag the **CheckOSMDbAdminRole** and drop it on the Action flow.

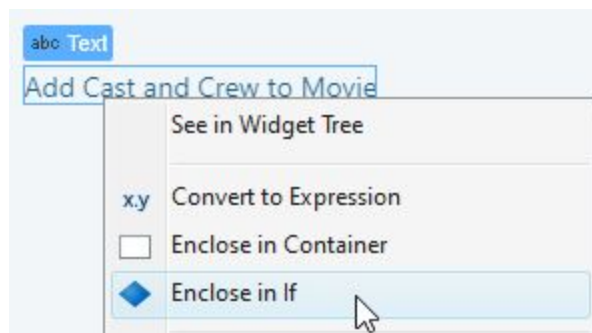


- d. Drag an **Assign** and drop it below the Check Action. Define the following assignment

HasRole = CheckOSMDBAdminRole.HasRole



- e. Open the MovieDetail Screen, right-click on the Add Cast and Crew to Movie and select **Enclose in If**



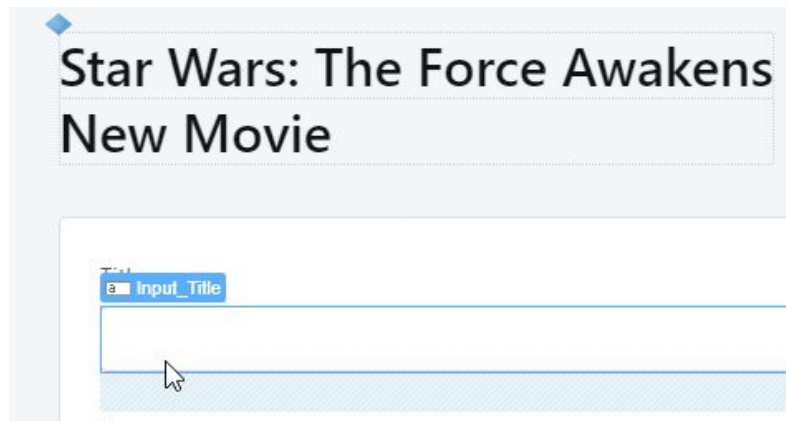
- f. Set the If **Condition** to *CheckAdminRole.HasRole*

- g. Publish the module and test it in the browser. Try to access the AddCastAndCrew anonymously. When you can't login with your OSMDbAdmin user and try again.
- h. Apply the same strategy to the **New Movie** link on the Movies Screen and to the **New Person** link on the People Screen.

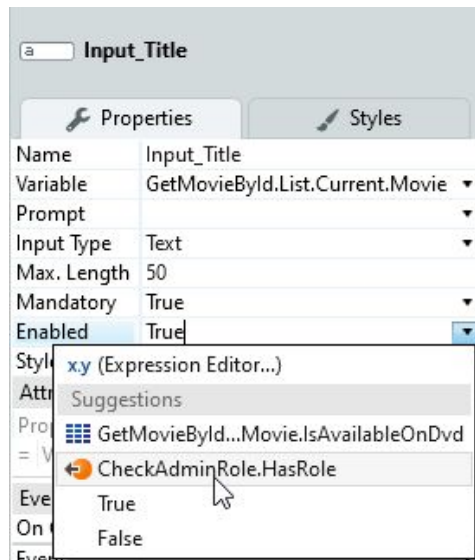
Disable Inputs in Screens

In this last section, we want to make sure that the information about movies and people can only be edited by OSMDbAdmin users. The other users can only see the info. For that, we will disable the input fields.

1. In the MovieDetail Screen, make sure that only OSMDbAdmin users can edit the information about the movie. For that, we want to disable the inputs and hide the save button.
 - a. Open the MovieDetail Screen and select the input field for the Title.

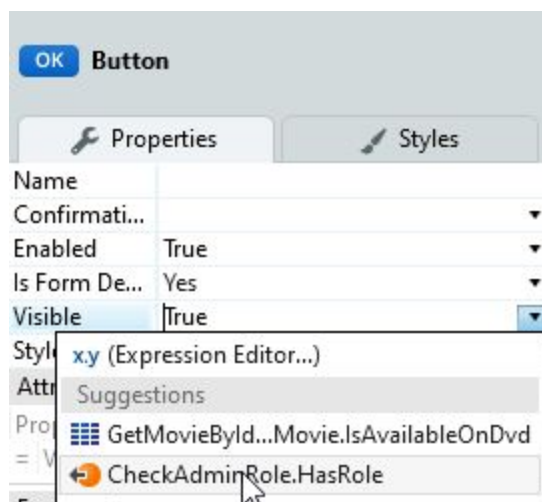


- b. In the properties of the Input, change the **Enabled** property to *CheckAdminRole.HasRole*



NOTE: This property defines in which circumstances the input field is enabled for editing. When set to True, it's always enabled for editing. In this case, it will only be enabled for users with the OSMDbAdmin role.

- c. Repeat the process for the remaining input fields.
- d. Select the Button and set the **Visible** property to *CheckAdminRole.HasRole*



- e. **Publish** and test your new features!
2. Apply the same strategy to the PersonDetail Screen.