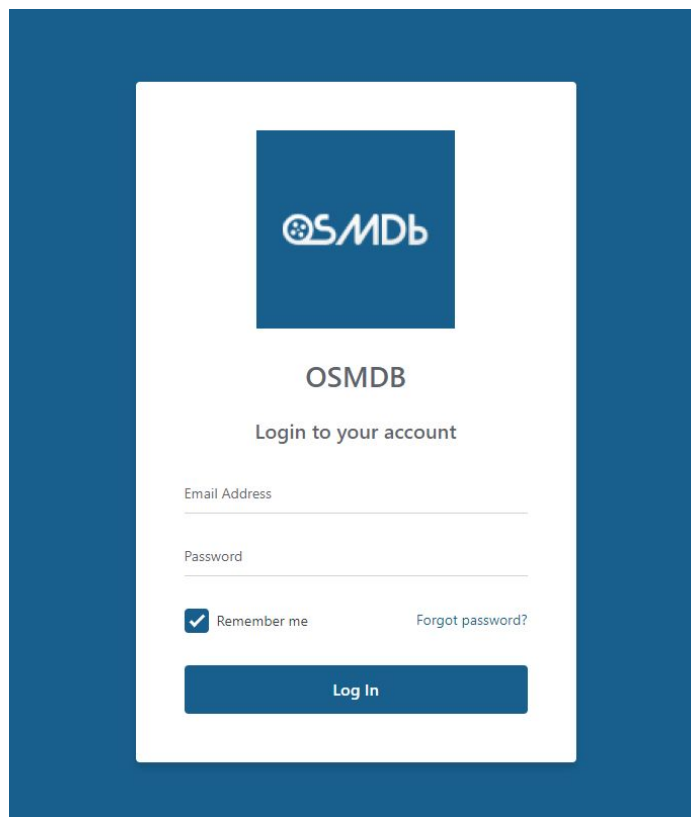


Security Exercise



The image shows a login form for OSMDB (OutSystems Managed Database) centered within a blue rectangular frame. The form itself has a white background. At the top of the form is the OSMDB logo, which consists of a blue square containing the text 'OSMDB' in white. Below the logo, the text 'OSMDB' is displayed in a bold, black font. Underneath that, the text 'Login to your account' is shown in a smaller, black font. The form contains two input fields: 'Email Address' and 'Password', both with light gray placeholder text. Below the 'Email Address' field is a checkbox labeled 'Remember me' with a checked mark, and to its right is a link labeled 'Forgot password?'. At the bottom of the form is a blue button with the text 'Log In' in white.

Introduction

We will now secure our application, making sure that only the people who are supposed to do some action, or access some Screen, are in fact able to do so. In particular, we want to distinguish between administrators' and registered users in the OSMDb application. The administrator users will have permissions to manage the information in the OSMDb's database, like add a new movie, edit the information of an existing movie, add a new person, among other operations. The registered users that are not administrators, will only be able to access the information provided.

OutSystems follows a Role-based access control approach to restrict, or allow, end-users to access specific Screens and operations in our applications. So, let's implement access control in the OSMDb application.

In this specific exercise lab, we will:

- Create Roles for an application
- Create users and grant them Roles
- Restrict access to parts of the application, to users with specific Roles
- Verify that a user has the privileges to access a specific part of the application

Table of Contents

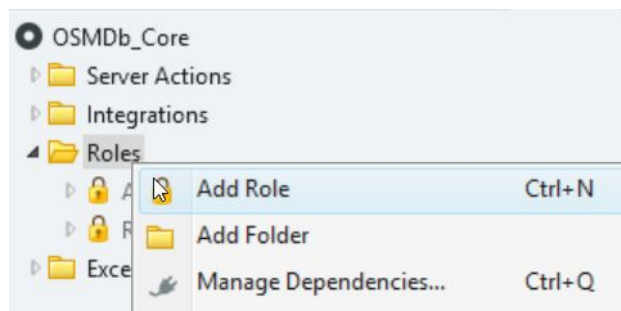
Introduction	2
Table of Contents	3
Create Roles and Grant them to Users	4
Restrict Access to Administration Areas	8
End of Lab	17

Create Roles and Grant them to Users

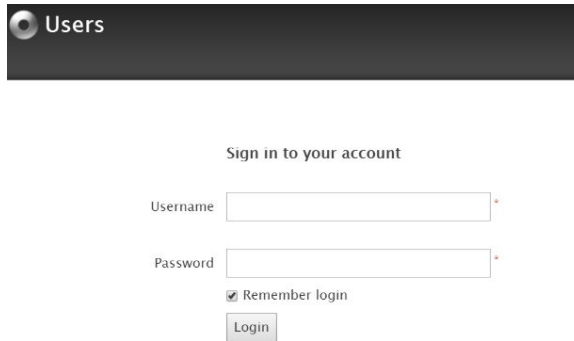
We will start by creating a new Role in the application, *OSMDbAdmin*. Roles can be granted to users and used to restrict access to parts of the application. Then, we will also create several users with different Roles, in the **Users** application. These users will have different permissions in the application, that will be defined by the Role they have.

This new Role will be created in the *OSMDb_Core* module, the Producer, since it will be used on both modules.

- 1) Create the new Role, *OSMDbAdmin*, in the **OSMDb_Core** module. This Role will be used to filter the access to certain Screens of the application. Make the Role **Public**.
 - a) Open the *OSMDb_Core* module.
 - b) Switch to the Logic tab, right-click over the folder **Roles** and select *Add Role*. Set its **Name** to *OSMDbAdmin*. Make the Role **Public**.



- c) Make the new Role **Public**.
 - d) Click the **1-Click Publish** button to publish the module.
- 2) Create a new user in the **Users** application and grant it the **OSMDbAdmin** Role. Add three more users. These users will have different permissions, taking into account the Role granted to them.
 - a) In your browser, access the application **Users**, using the URL:
https://<your_server>/Users
 - b) If you are already logged in, you will access the **Users** application. Otherwise, you will get to a **Login** page like in the following screenshot



Users

Sign in to your account

Username

Password

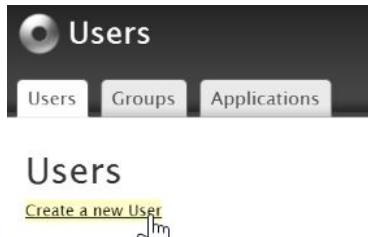
☒ Remember login

Login

- c) If you need to Login, use the Username: **admin** and Password: **outsystems**.

NOTE: If you are following the **Online Class**, using your Personal or Enterprise Environment, you should login with your credentials for logging in to the Environment.

- d) In the application **Users**, select the option *Create a new User*, to open the registration form.




Users

Users Groups Applications

Users

Create a new User

- e) Fill the form to create a user **movieadmin_<YOUR_INITIALS>** with password: *outsystems*. Set its **Name** to *movie admin*. Press **Save** to create the user. From simplicity purposes, this user will be referenced as *movieadmin* from now on.



Name

Username

Email

Phone

Save or Cancel

- f) You will be redirected to the page of the recently created user, **movieadmin**.

movie admin

[Edit this User](#)

Username movieadmin
 Email movieadmin@osmdb.com
 Phone
 Creation Date 19 Oct
 Last Login 16:37 (17 minutes ago)

Groups

Type or double-click for list	Add
-------------------------------	-----

Roles

Type or double-click for list	Add
-------------------------------	-----

- g) In the **Roles** section, double-click the input box, select *OSMDBAdmin* and click **Add**, to grant the **OSMDBAdmin** Role to the user.

Roles

OSMDBAdmin (OSMDB_Core)	Add
-------------------------	-----

NOTE: The module's name appears next to the Role's name, within parenthesis. This information depends on the name given to the module. For instance, if the module name is **OSMDB_abc**, then the option would appear as *OSMDBAdmin (OSMDB_abc)*.

- h) You should see the granted Role, under the **Roles** section in the user's page.
 i) Go back to the **Users** main page in the application.

Users

[Create a new User](#)

<input type="text"/>	Search
----------------------	--------

 movie admin movieadmin@osmdb.com	movieadmin
--	------------

- j) Following the same steps than before, create three new users, e.g: *Mary Jane*, *John Smith* and *Jack Daniels*. Do not assign them a Role, and use a password of your choice. If you're following a classroom training, don't forget to add your initials to uniquely identify your user.

Mary Jane

[Edit this User](#)

Username mary.jane
Email mary.jane@osmdb.com
Phone
Creation Date 20:52 (just now)
Last Login

Groups

Type or double-click for list

Roles

Type or double-click for list

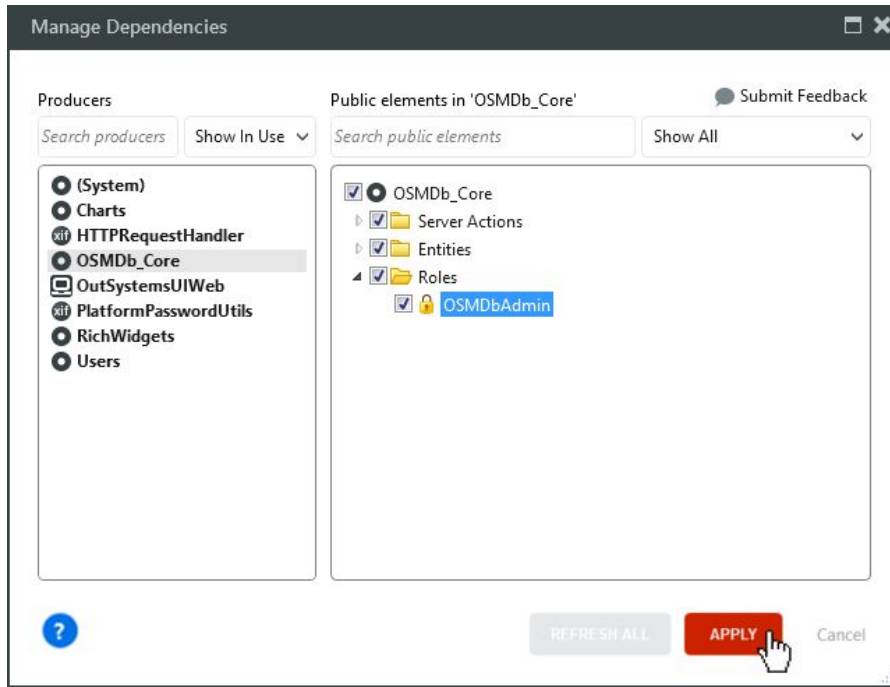
NOTE: The **Registered** Role is automatically granted to any User created in the **Users** application, even without explicitly assigning a Role to the User.

If you are using the development environment with other students of the course, the Users created should have your initials, or just create your own names, to uniquely identify them. It is possible to create several Users with the same data, however with other users in the same environment you will see, for instance, all Mary Janes created by the other users.

Restrict Access to Administration Areas

Let's use the **OSMdbAdmin** Role to restrict parts of the application, only to users that have this Role. We will restrict the access to Screens and to specific functionalities of the application, using an Action automatically created with the new Role.

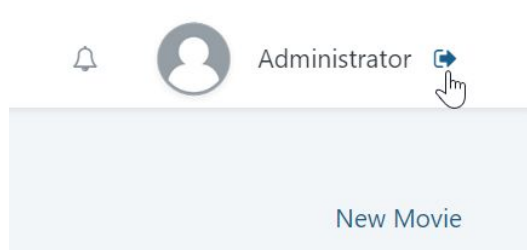
- 1) We will need to use the OSMdbAdmin Role in the **OSMdb** module. Open the module and add a reference to the **OSMdbAdmin** Role, under the OSMdb_Core module.



- 2) Restrict the access to the **AddMovieParticipant** Screen exclusively to users with the Role **OSMdbAdmin** assigned to them.
 - a) In the MainFlow, select the **AddMovieParticipant** Screen and untick all the Roles, except for the **OSMdbAdmin**. After this step, the properties of the Screen should be like this

<input type="checkbox"/>	AddMovieParticipant Web Screen
Name	AddMovieParticipant
Description	...
Public	No
Title	
Roles	
Anonymous	<input type="checkbox"/>
Registered	<input type="checkbox"/>
OSMdbAdmin	<input checked="" type="checkbox"/>

- b) Click the **1-Click Publish** button to publish the application, and access it using your browser.
- c) If any user is logged in, make sure you log out the application.



- d) In the Movies Screen, select a movie and click on the option to **Add Cast/Crew to Movie**.
- e) You will get to a Login page. Login with the **OSMDBAdmin** credentials.

Username: *movieadmin*

Password: *outsystems*

OSMDB

Login to your account

Email Address

Password



Remember me

[Forgot password?](#)

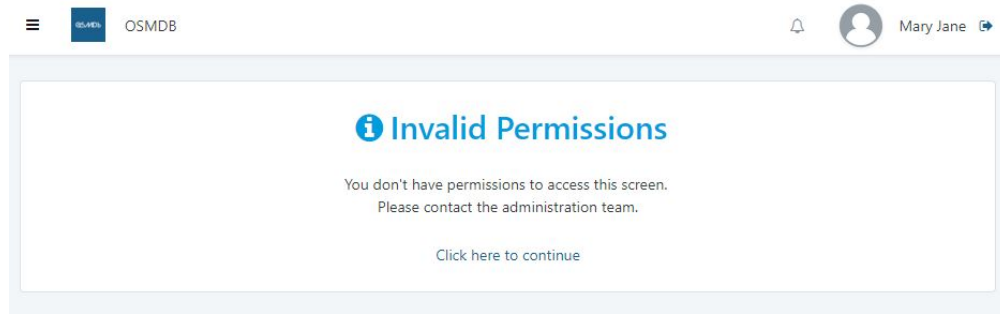
Log In

- f) You will get to the **AddMovieParticipant** Screen and will see on the top right corner the User that is logged in.



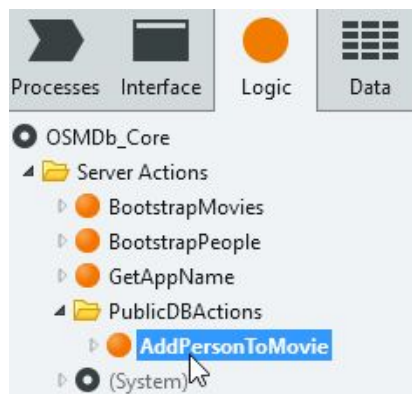
- g) Logout with the user **movieadmin**, to be redirected to the Login page again.
- h) Login with the user *Mary Jane*, select a movie and click **Add Cast/Crew to Movie**.

- i) You will end in an error page, indicating *Invalid Permissions*, as Mary Jane does not have the **OSMDBAdmin** Role.

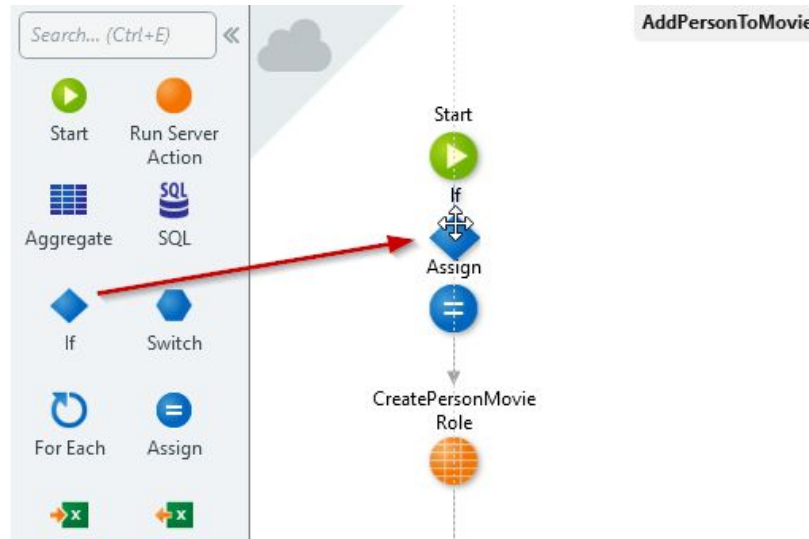


NOTE: The **AddMovieParticipant** Screen will only be accessible by users with the **OSMDBAdmin** Role. All the other Screens in the application have the Anonymous Role ticked, meaning that everyone can access it, even without being Registered or without having a Role associated.

- 3) We restricted the access to the **AddMovieParticipant** Screen at the level of the UI. However, it is an OutSystems Best Practice to also check the Role in the business logic as much as possible, even if the UI apparently restricts the access. For that reason, go back to the **AddPersonToMovie** Action in the **OSMDB_Core** and don't allow a user without the **OSMDBAdmin** Role to add a person to a movie. If the user does not have that Role, it should Raise an Exception indicating that the user does not have the Role.
- a) Switch to the **OSMDB_Core** module and open the **AddPersonToMovie** Server Action under the **PublicDBActions** folder.



- b) Drag an If statement just after the Start node of the Action flow.



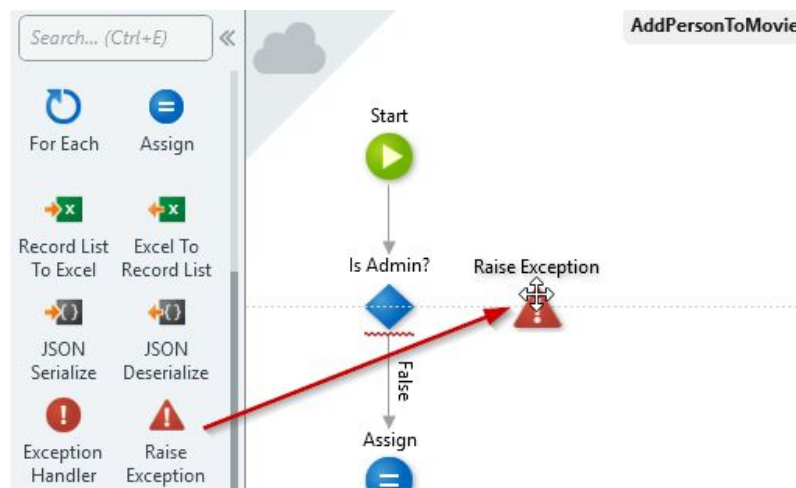
- c) Set the Condition of the If to:

CheckOSMDbAdminRole()

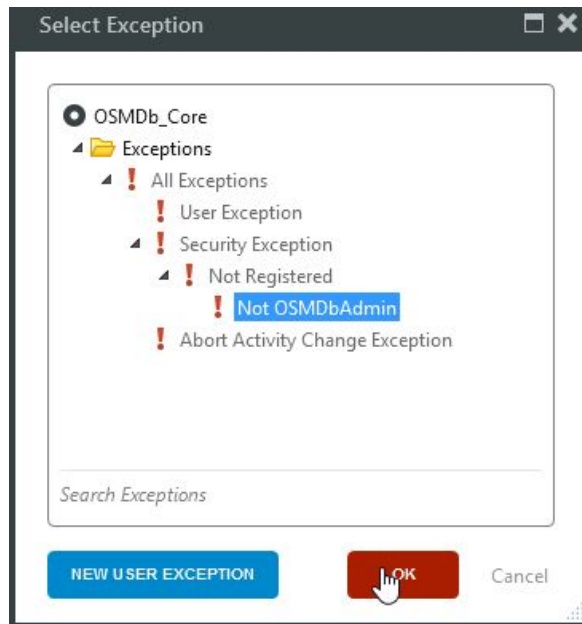
NOTE: The **Check<RoleName>Role()** Action determines if the user currently logged in has the *RoleName* Role. This Action has an optional input parameter that expects the User Id. If left empty, assumes the user currently logged in.

All Roles have this Action available, including the **Registered** Role.

- d) Set the **Label** of the If to *Is Admin?*
- e) Drag and drop a new Raise Exception next to the If.



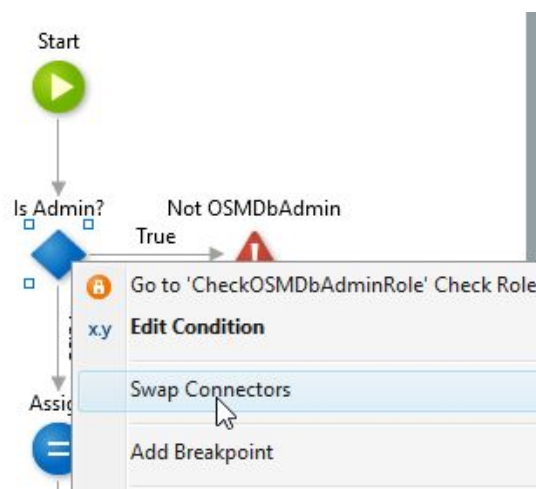
- f) In the new dialog, select the *Not OSMDbAdmin* Exception. This Exception is under the **Security Exceptions**, meaning that it can be handled by a Security Exception Handler.



- g) Set the Exception Message to:

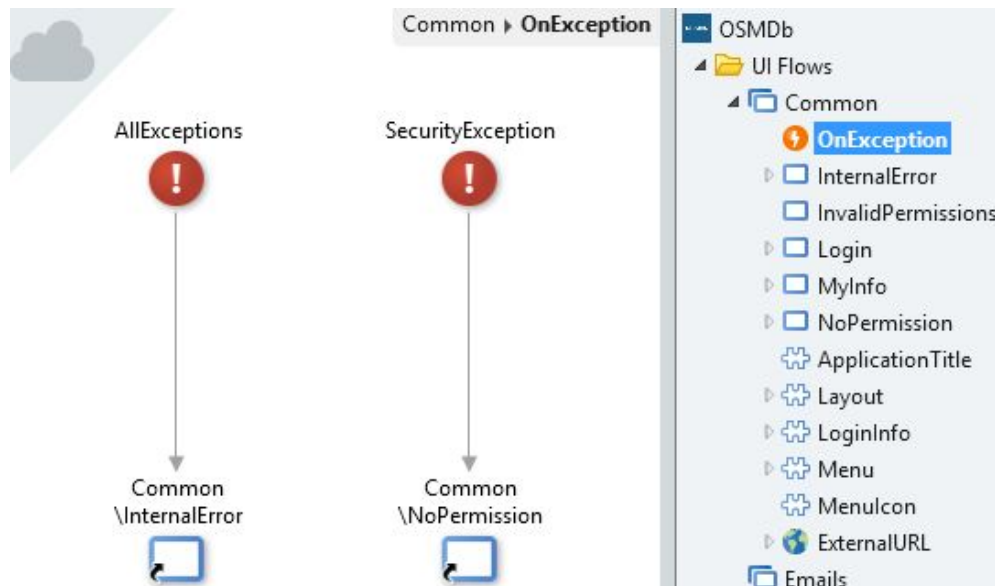
"User does not have permissions"

- h) Connect the If to the Raise Exception. Notice that it created the **True** branch. However, if the user is OSMDbAdmin, we want it to proceed the logic to add the record and not to Raise an Exception. Right-click on the If and choose *Swap Connectors*.

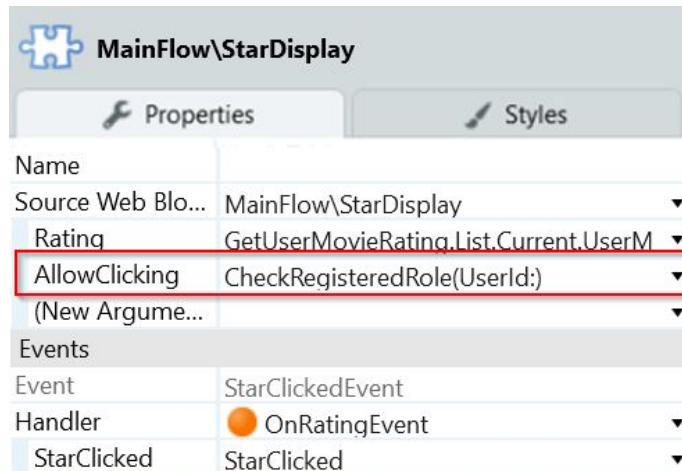


- i) Publish the OSMDb_Core module.

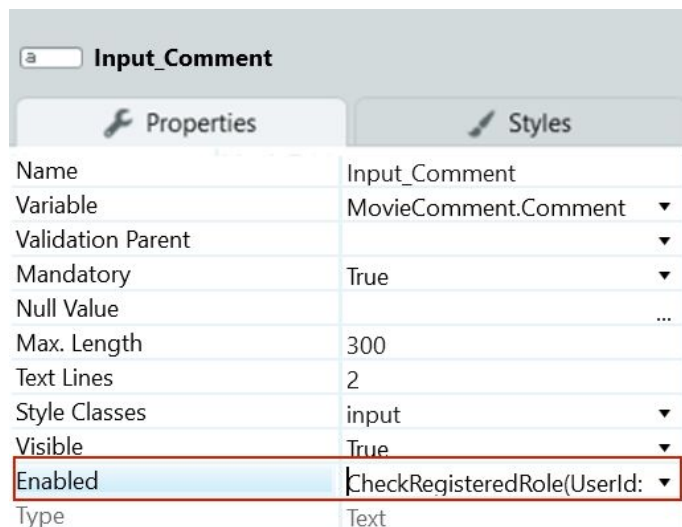
- j) Switch to the OSMDb module and click on F5 to publish the module. Since there are no breaking changes in the **AddPersonToMovie** Action, publishing the consumer is enough to make it use the new version of the Action. You might have noticed that we didn't create an Exception Handler for the new Raise Exception. However, this Exception is handled by the Global Exception Handler. In the Common UI Flow, under the Interface tab, there is the OnException element that has a Security Exception Handler, that is able to handle the **Not OSMDbAdmin** Exception, if raised.



- 4) Restrict the comment section of the **MovieDetail** Screen. Everyone that accesses the application is able to view movie comments and the ratings, but only a **Registered** user can rate a movie, or add a new comment.
- Open the **MovieDetail** Screen and select the **StarDisplay** Web Block in the sidebar of the Screen.
 - Change the value of the **Allow Clicking** parameter, to check if the user logged in is **Registered**, with *CheckRegisteredRole(UserId:)*. This is the exact same representation of *CheckRegisteredRole()*, meaning that no value to the Input Parameter is being passed.



- c) Select the **Input_Comment** box and set the **Enabled** property to *CheckRegisteredRole(UserId:)*

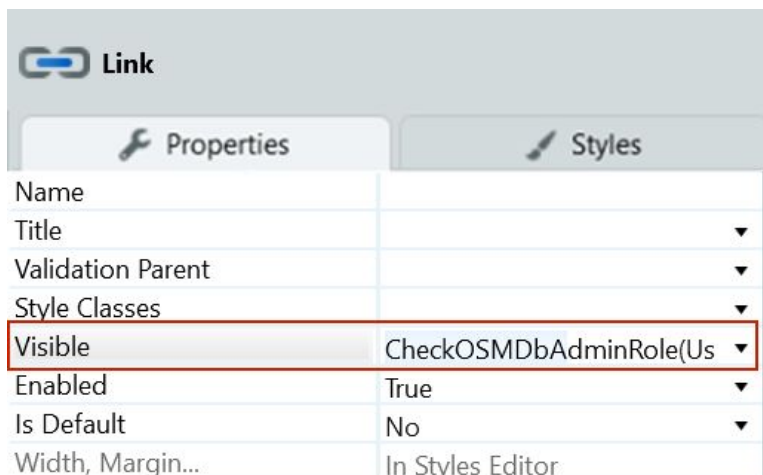


- d) Select the **Comment** Button and repeat the previous step.

NOTE: The **Enabled** property defines if, or when, the respective Widget is available to be used. In this particular case, if a user is not **Registered** it cannot write on the **Input_Comment** box, press the **Comment** Button, or add a new rating, despite seeing those options on the page.

- 5) Click the **1-Click Publish** button to publish the module, and access it using your browser to test the application.
- a) If any user is logged in, make sure you log out the application. In your browser, access the OSMDb application, using the URL: *https://<your_server>/Movies.aspx*

- b) Select a movie in the **Movies** Screen and check that the options to rate a movie and add a comment are disabled (e.g: you cannot add a new comment or rate a movie).
 - c) Login with any created user, e.g: Mary Jane.
 - d) Repeat step **b)** and check that the options to rate the movie and add a new comment are available. Try to rate a movie.
- 6) Restrict some functionalities of the application to users with the **OSMDBAdmin** Role. So far, we restricted the **AddMovieParticipant** Screen to be accessible only by OSMDBAdmin users. Also, we added logic to the **AddPersonToMovie** Action, to avoid that a user without this Role is able to add a person to a movie. However, there is still a Link in the MovieDetail Screen available to everyone to click. So, it makes sense that only users with this Role will be able to see the **Add Cast/Crew to Movie** Link. Also, we will only allow OSMDBAdmins to edit the information about a movie and artist, and add new movies and artists.
- a) Open the **MovieDetail** Screen.
 - b) Select the **Add Cast/Crew to Movie** Link and change the **Visible** Property to *CheckOSMDBAdminRole(UserId:)*



Link	
Properties	Styles
Name	
Title	▼
Validation Parent	▼
Style Classes	▼
Visible	CheckOSMDBAdminRole(Us ▼
Enabled	True ▼
Is Default	No ▼
Width, Margin...	In Styles Editor

NOTE: Using the **Visible** property of a Link / Button, one can define the condition for a Link / Button to appear in the page. By default, this property is set to 'True', meaning that it always appears.

- c) Select the **MovieForm** and change the **Enabled** property to *CheckOSMDBAdminRole(UserId:)*

NOTE: In a Form, the **Enabled** property defines in which circumstances the Form is editable. By default, this property is set to True, meaning that the Form is always editable.

- d) Select the **Save** Button and change its **Visible** property to *CheckOSMDbAdminRole(UserId:)*
 - e) Open the **PersonDetail** Screen.
 - f) Select the **PersonForm** and change its **Enabled** property to *CheckOSMDbAdminRole(UserId:)*
 - g) Select the **Save** Button and change its **Visible** property to *CheckOSMDbAdminRole(UserId:)*
 - h) Open the **Movies** Screen.
 - i) Select the **New Movie** Link and change its **Visible** property to *CheckOSMDbAdminRole(UserId:)*
 - j) Repeat the same logic for the **People** Screen and the **New Person** Link.
- 7) Publish the module and test the application using the user Mary Jane and movieadmin, to understand the differences and make sure that it works as expected.
- a) Click the **1-Click Publish** button to publish the application, and access it using your browser.
 - b) Login as *Mary Jane*.
 - c) Navigate to the **Movies** Screen and check that the **New Movie** Link is missing in the Actions placeholder.
 - d) Select a movie and check that it is not possible to edit its information in the Form.
 - e) Login with the user *movieadmin*.
 - f) Repeat the previous steps, and check that the **New Movie** Link is visible and that the information about the movie is editable.

End of Lab

In this exercise lab, we have learned how to create new Roles in an application and how to create users, and grant them roles, using the **Users** application. We created a new Role called OSMDbAdmin and granted it to one new user (movieadmin).

Then, we have learned how to restrict Screens and functionalities of the application to users with certain Roles. In a Screen, we can select the users that have permissions to access it, by ticking the Roles in the Screen properties.

Roles also have an Action/Function associated to them, **Check<RoleName>Role()**, that determines if a user has the Role **<RoleName>**. This Action can be used in the Widgets or other Actions, to define some additional logic regarding the permissions of the users, to a particular functionality of the application. We used this Action for the OSMDbAdmin Role, to disable Forms and Buttons, to hide Links and also to add business logic to the **AddPersonToMovie** Action, to make sure only OSMDbAdmins could perform this operation.