outsystems

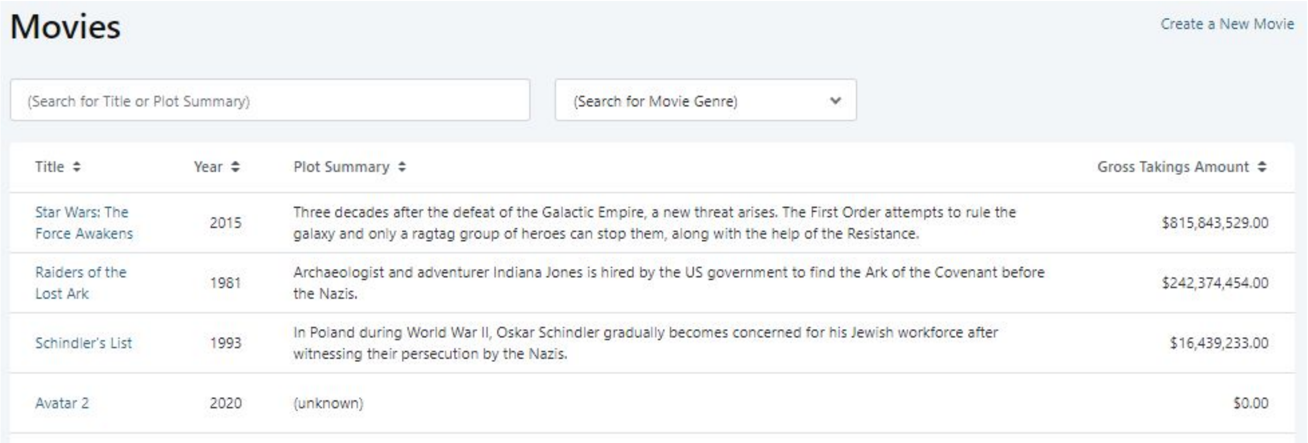
Search Filters - How-to

Table of Contents

Outline

In this exercise, we will use Client Variables to expand the functionality of the application.

This will be done by adding a filter capability on the Movies Screen. We want to be able to search by movie title, plot summary and movie genre and we want to make sure that when we change to a different Screen, and come back, the search filter was not lost. The search filter should look like this:



Movies

Create a New Movie

(Search for Title or Plot Summary)

(Search for Movie Genre)

Title	Year	Plot Summary	Gross Takings Amount
Star Wars: The Force Awakens	2015	Three decades after the defeat of the Galactic Empire, a new threat arises. The First Order attempts to rule the galaxy and only a ragtag group of heroes can stop them, along with the help of the Resistance.	$815,843,529.00
Raiders of the Lost Ark	1981	Archaeologist and adventurer Indiana Jones is hired by the US government to find the Ark of the Covenant before the Nazis.	$242,374,454.00
Schindler's List	1993	In Poland during World War II, Oskar Schindler gradually becomes concerned for his Jewish workforce after witnessing their persecution by the Nazis.	$16,439,233.00
Avatar 2	2020	(unknown)	$0.00

Note that the search should be done automatically, when the users finished typing the title/plot summary keyword, or when it just chose the movie genre. Also, the end-user does not need to type the whole movie title and the whole plot summary, for the search to return the movies that match the keyword that was typed.

How to

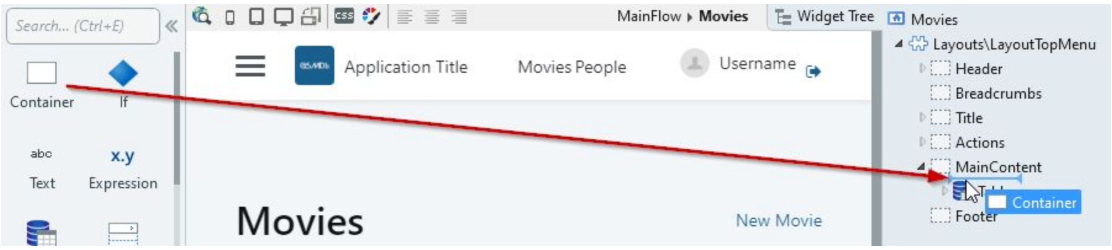
In this section, we'll describe, step by step, the exercise *8.3 - Search Filters*.

Adding Search Input Fields

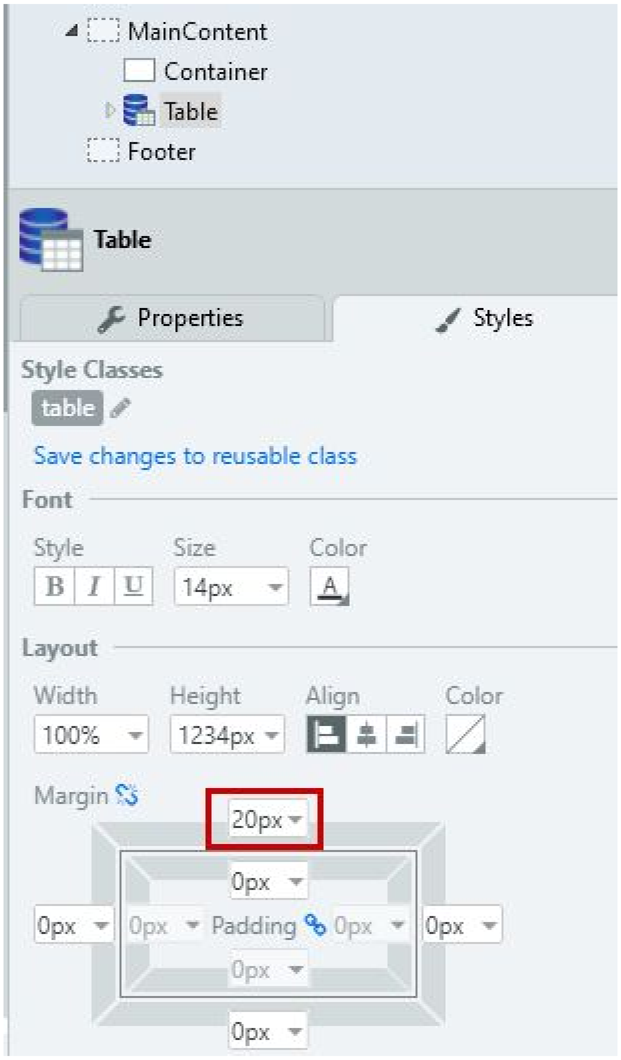
To allow searching for movies by title/plot summary and movie genre, we need to add input fields to the Screen to allow the end-user to define the desired search filters.

1. Drag two input widgets, one Input and one Dropdown to the top of the movies table.

a. Start by finding space on the Screen for the search widgets. In the Movies Screen, open the **Widget Tree**, expand the **MainContent** and drag a **Container** right above the Table.

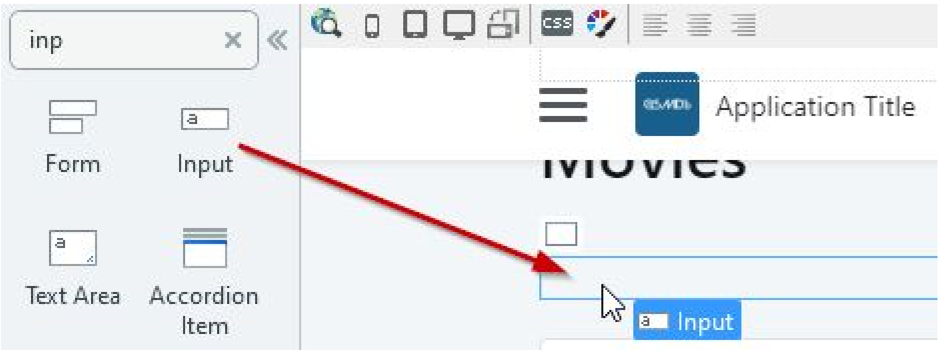


b. Select the Table, switch from the Properties to the **Styles** view and add a margin top of *20px*

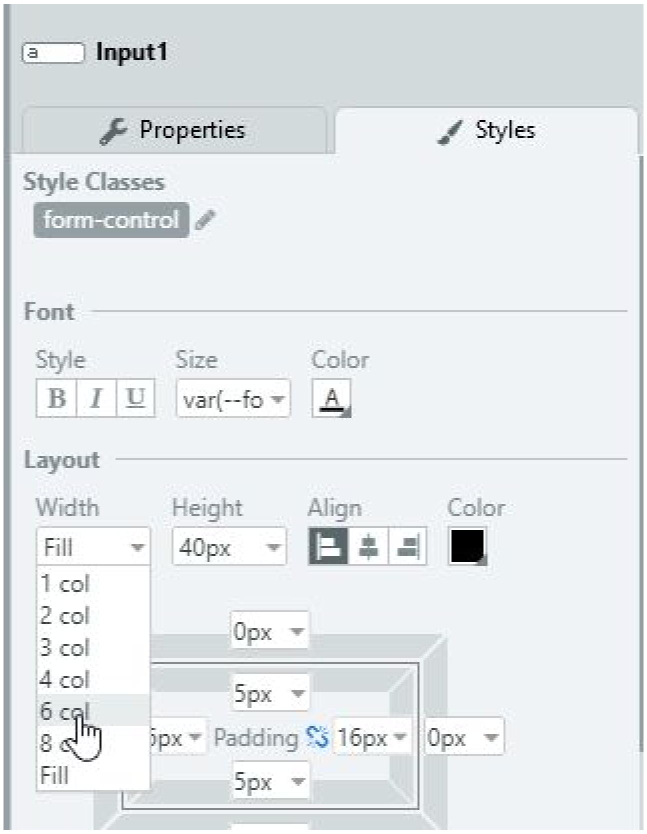


Style: B I U
Size: 14px
Width: 100%
Height: 1234px
Margin: 20px (top), 0px (left), 0px (right), 0px (bottom)
Padding: 0px, 0px, 0px, 0px

c. Drag an **Input** widget and drop it inside the Container



d. Set the **Width** of the Input to *6 columns*



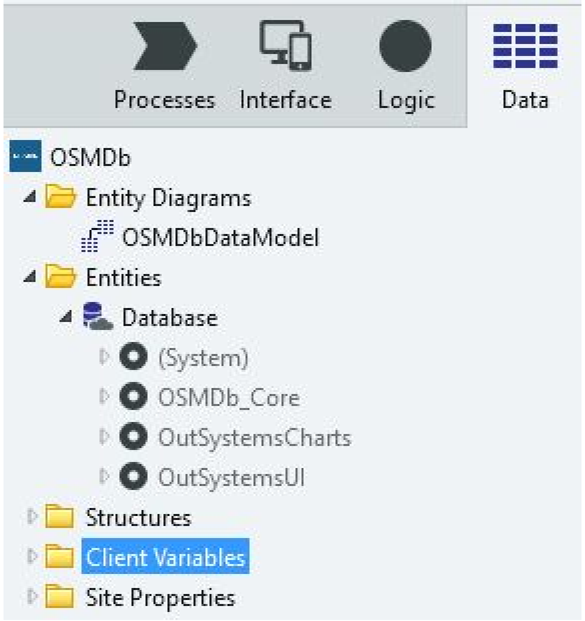
e. Drag a **Dropdown** widget and drop it right next to the Input widget

f. Set its **Width** to *4 columns*

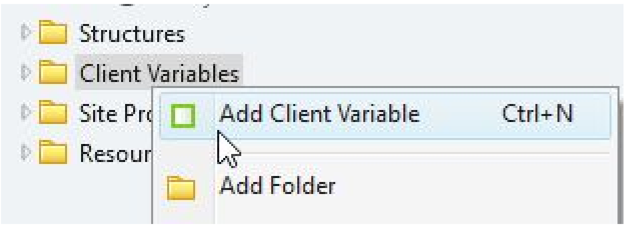
g. At this point, the two widgets have errors, which will be fixed in the later steps.

2. Now that we have the two input fields on the Screen, we need to define the variables where the searched filters, meaning the values that the end-users insert in the input fields, are saved. We will use Client Variables to make sure that the search filter is not lost when we change to a different Screen and come back to the Movies Screen.

    a. Switch to the Data tab in Service Studio and locate the **Client Variables** folder.

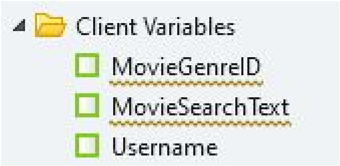


    b. Right-click on the Client Variables folder and select **Add Client Variable**

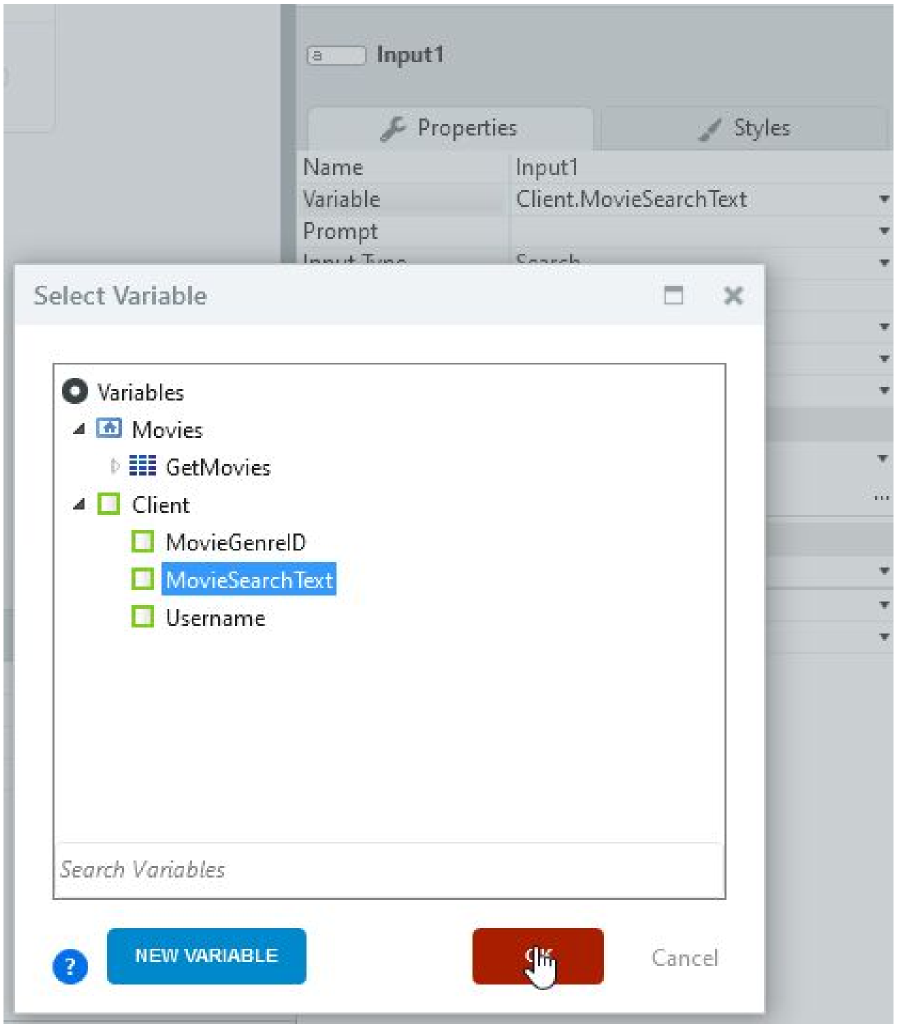


    c. Set the **Name** of the Variable to *MovieSearchText* and make sure the **Data Type** is set to *Text*.

    d. Repeat the previous two steps, but this time for a Client Variable named *MovieGenreId* of **DataType** *MovieGenre Identifier*.



e. Go back to the Movies Screen, select the Input field and set the **Variable** property to the *MovieSearchText*

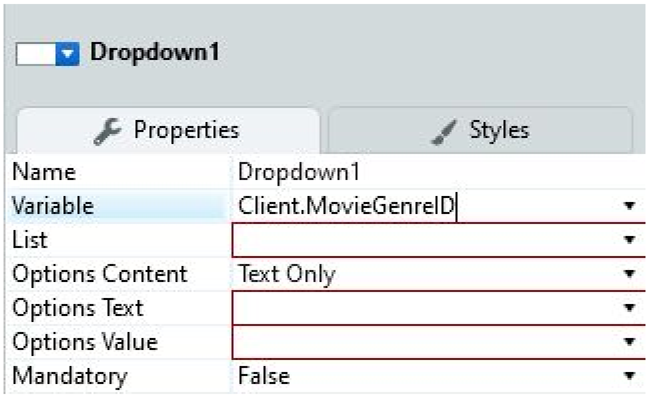


Name	Input1
Variable	Client.MovieSearchText
Prompt	
Input Type	Search

- Variables
  - Movies
    - GetMovies
  - Client
    - MovieGenreID
    - MovieSearchText
    - Username

Search Variables

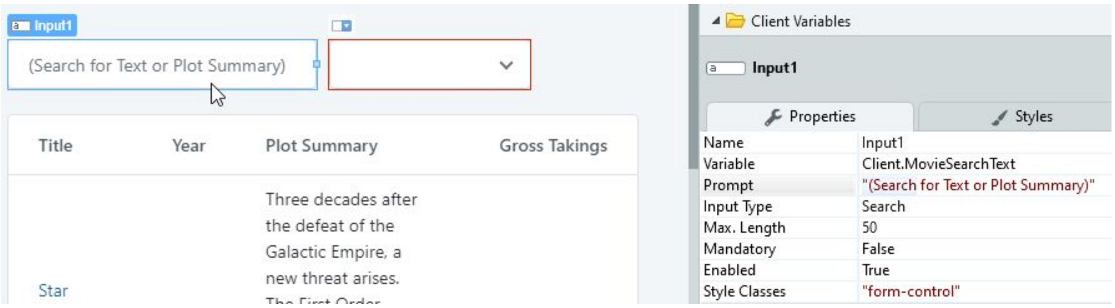
f. Select the Dropdown and set the **Variable** property to *Client.MovieGenreId*



Name	Dropdown1
Variable	Client.MovieGenreID
List	
Options Content	Text Only
Options Text	
Options Value	
Mandatory	False

3. Now we just need to finish the implementation of the input widgets, to remove the errors, but also to help the end-user understand what are the allowed search filters.

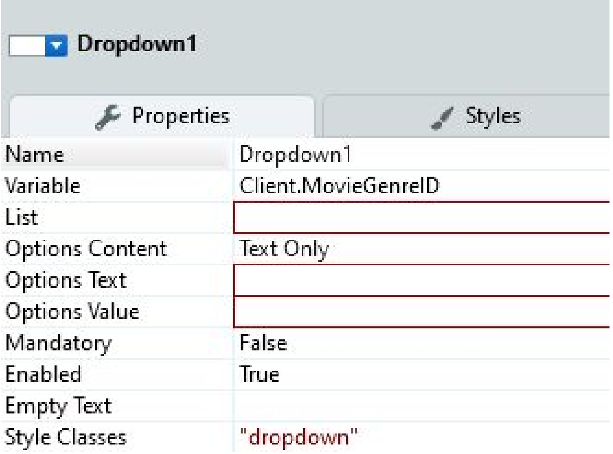
    a. Select the Input widget and set the **Prompt** property to *"(Search for Text or Plot Summary)"*



Input1
(Search for Text or Plot Summary)
Title
Year
Plot Summary
Gross Takings
Three decades after the defeat of the Galactic Empire, a new threat arises. The First Order
Star
Client Variables
Input1
Properties
Styles

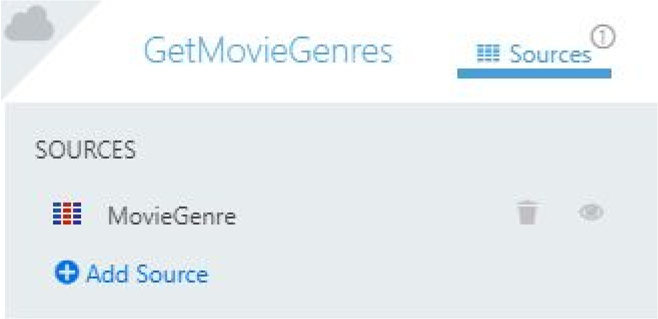
Name	Input1
Variable	Client.MovieSearchText
Prompt	"(Search for Text or Plot Summary)"
Input Type	Search
Max. Length	50
Mandatory	False
Enabled	True
Style Classes	"form-control"

    b. Select the Dropdown widget and notice that there are several errors we need to solve. The first one is to set what would be the List of values that the dropdown will display.

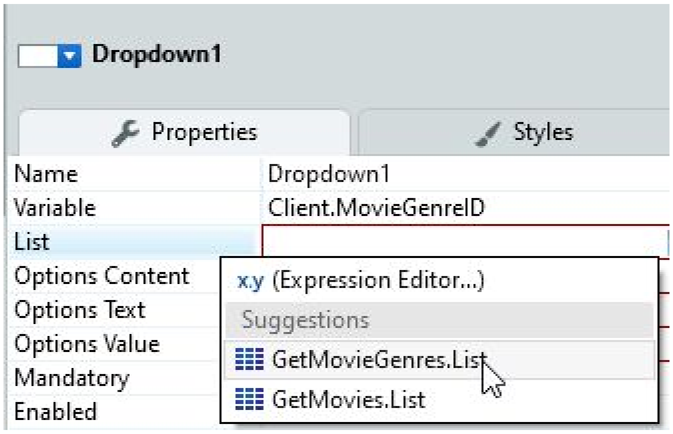


Name	Dropdown1
Variable	Client.MovieGenreID
List	
Options Content	Text Only
Options Text	
Options Value	
Mandatory	False
Enabled	True
Empty Text	
Style Classes	"dropdown"

    c. Add an **Aggregate** to the Screen and select the **MovieGenre** Entity as the single Source.

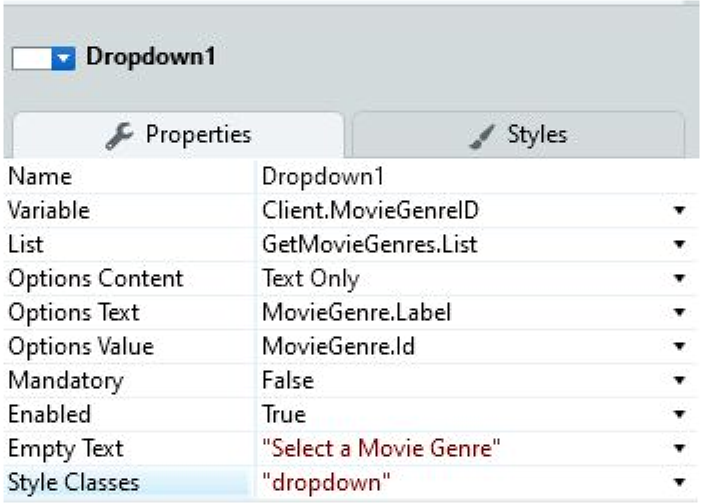


d. Set the **List** property of the Dropdown widget to the output of the Aggregate: *GetMovieGenres.List*



Name	Dropdown1
Variable	Client.MovieGenreID
List	
Options Content	
Options Text	
Options Value	
Mandatory	
Enabled	

e. All the errors are now fixed automatically. To finish, set the **Empty Text** to *"Select a Movie Genre"*



Name	Dropdown1
Variable	Client.MovieGenreID
List	GetMovieGenres.List
Options Content	Text Only
Options Text	MovieGenre.Label
Options Value	MovieGenre.Id
Mandatory	False
Enabled	True
Empty Text	"Select a Movie Genre"
Style Classes	"dropdown"

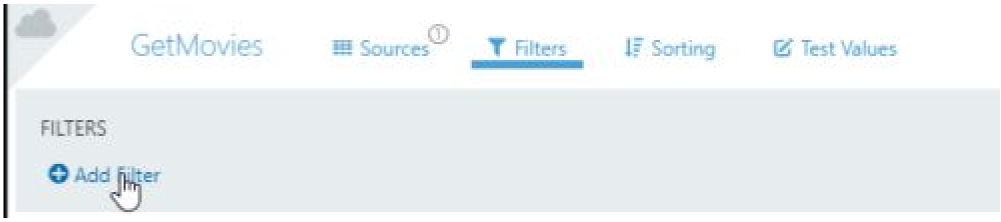
Applying the Search

The input fields are created on the Movies Screen. Now, we need to define the logic to actually apply the search on the Movies table. This has two steps: filter the GetMovies Aggregate by the search filters; triggering the search whenever the user selects the search filter.

1. In the GetMovies Aggregate, we need to define two additional filters, so that the movies can be filtered by the search criteria chosen by the end-users.

   a. Open the **GetMovies** aggregate.

b. Select the **Filters** tab and click on **Add Filter**



c. Add a filter condition for the title and plot summary

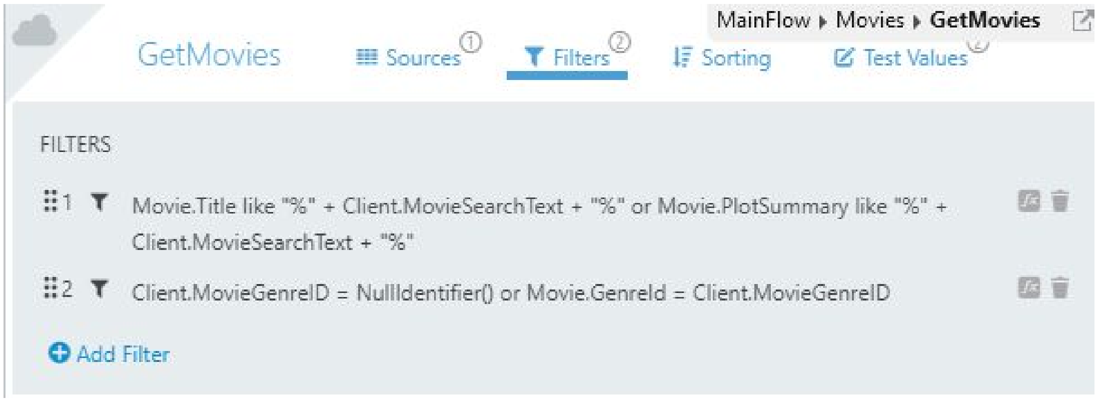
Movie.Title like "%" + Client.MovieSearchText + "%" or Movie.PlotSummary like "%" + Client.MovieSearchText + "%"

NOTE: The % are used as wildcards, so this expression can be read as: we want a movie title with some text (possibly empty), followed by the keyword and then another text (possibly empty). The same logic is valid for the plot summary.

---

d. Add a second filter for the **Movie Genre**

Client.MovieGenreId = NullIdentifier() or Movie.GenreId = Client.MovieGenreId



1 Movie.Title like "%" + Client.MovieSearchText + "%" or Movie.PlotSummary like "%" + Client.MovieSearchText + "%"

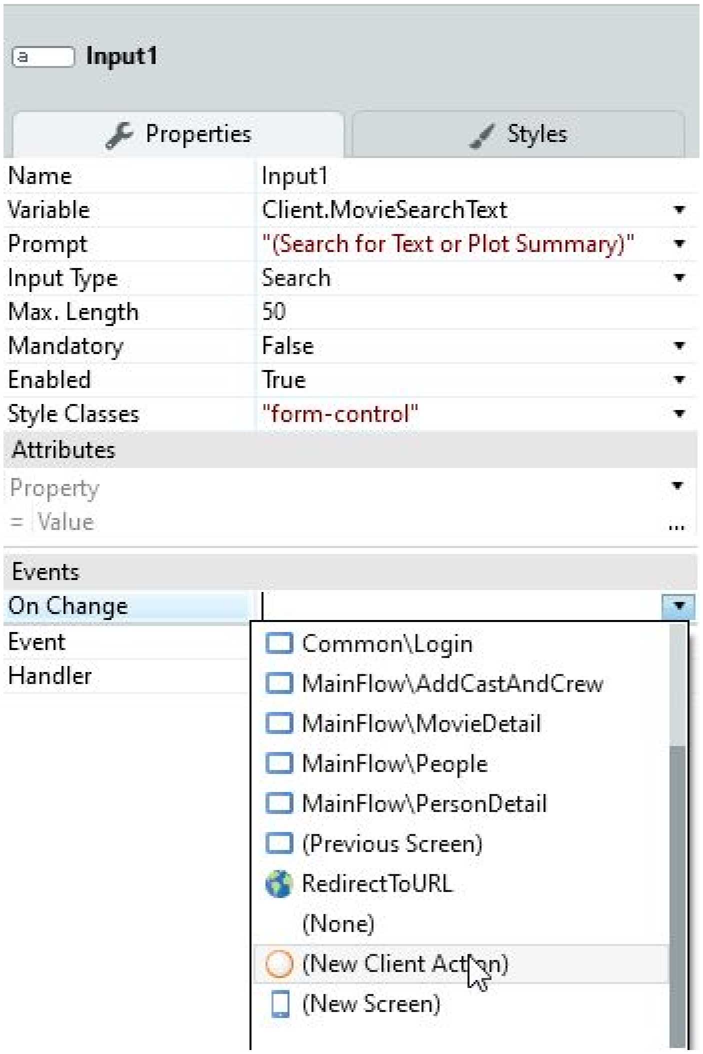
2 Client.MovieGenreID = NullIdentifier() or Movie.GenreId = Client.MovieGenreID

NOTE: If the condition was just *Movie.GenreId = Client.MovieGenreId,* whenever the user did not select a genre to filter the movies from, meaning that all the genres should be displayed, no results would be returned. This happens because there are no records in the database, where the Movie.GenreId would be NullIdentifier() (no option chosen). So, we need to consider that scenario as well on the Aggregate filter.

---

2. Now, we need to trigger the search, whenever the user types a title or plot summary, or selects a movie genre in the dropdown. For that, we need to use the OnChange Event.

    a. Select the Input field on the Movies Screen and set the **OnChange** property, under **Events**, with a *(New Client Action)*

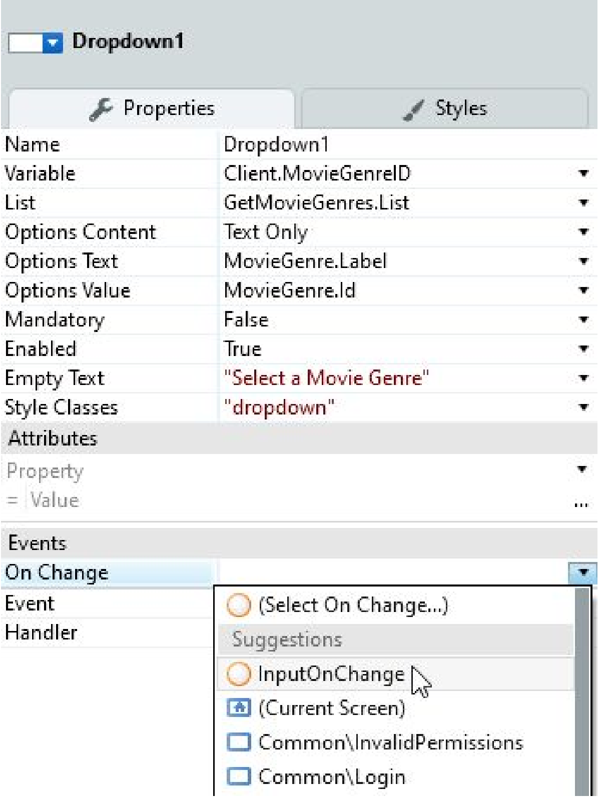


Input1

Name	Input1
Variable	Client.MovieSearchText
Prompt	"(Search for Text or Plot Summary)"
Input Type	Search
Max. Length	50
Mandatory	False
Enabled	True
Style Classes	"form-control"
Attributes	
Property	
= Value	...
Events	
On Change	
Event	
Handler	

    b. Rename the generated Client Action to *InputOnChange*

c. Set the **OnChange** Event of the Dropdown to the same Action.



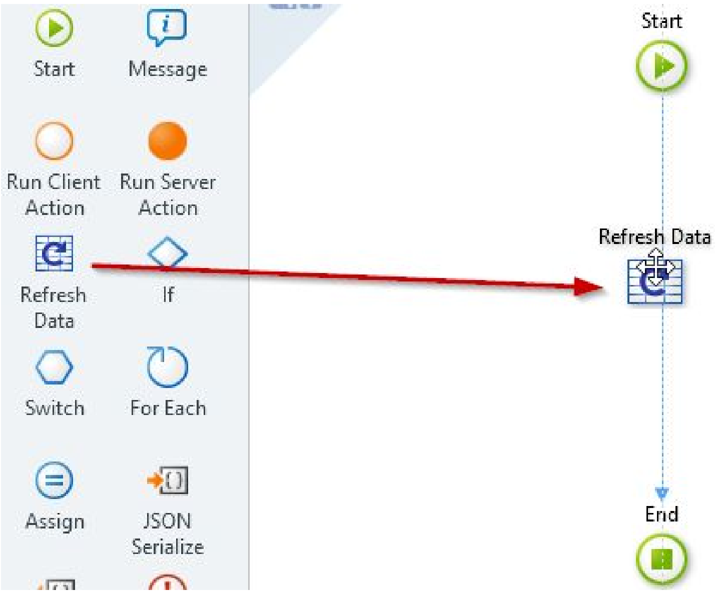
Dropdown1

Properties | Styles

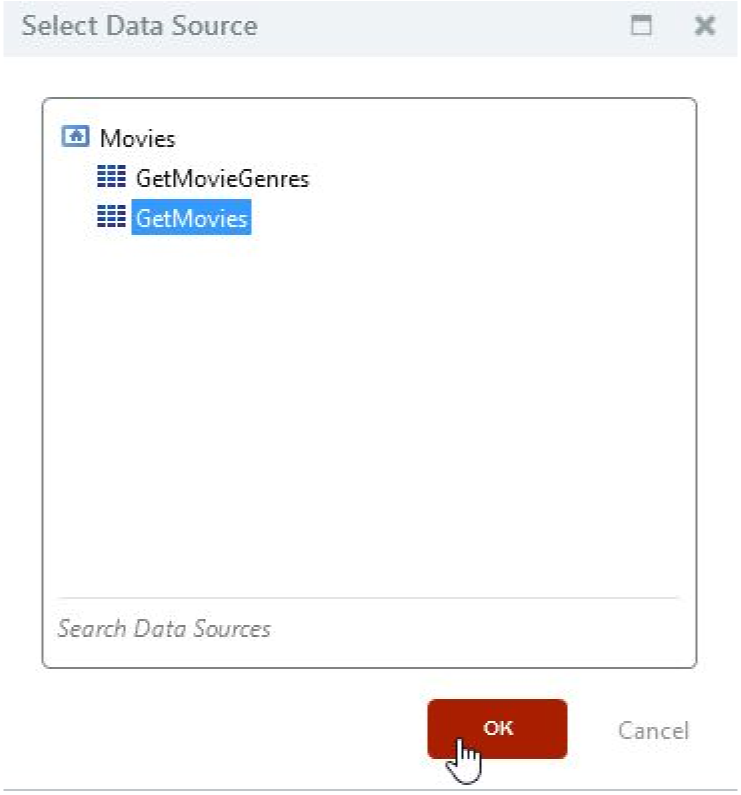
Name	Dropdown1
Variable	Client.MovieGenreID
List	GetMovieGenres.List
Options Content	Text Only
Options Value	MovieGenre.Id
Options Text	MovieGenre.Label
Mandatory	False
Enabled	True
Empty Text	"Select a Movie Genre"
Style Classes	"dropdown"
Attributes	
Property	
= Value	...
Events	
On Change	
Event	
Handler	

(Select On Change...)
Suggestions
InputOnChange
(Current Screen)
Common\InvalidPermissions
Common\Login

d. Open the **InputOnChange** Action, drag a **RefreshData** and drop it in the Action Flow.



e. Select the **GetMovies** Aggregate in the new dialog.



NOTE: With the OnChange Event, as soon as the end-user stops typing, or selects a dropdown option, the Client Action is automatically triggered. In this particular example, we just need to refresh the GetMovies Aggregate, so that it runs again and applies the search filters to the list of movies. The UI is automatically updated.

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