Blocks and Events - Exercise

Table of Contents

Outline

In this exercise we will introduce Blocks in our application, to allow a user to rate a movie. That rating will be based on a star rating system, where the user can rate a movie from 1 to 5 stars. This rating functionality should be added to the MovieDetail Screen.

Also in the same Screen, we should have the information of the Average Rating of the movie, which will combine all the ratings given by the end-users to that particular movie.

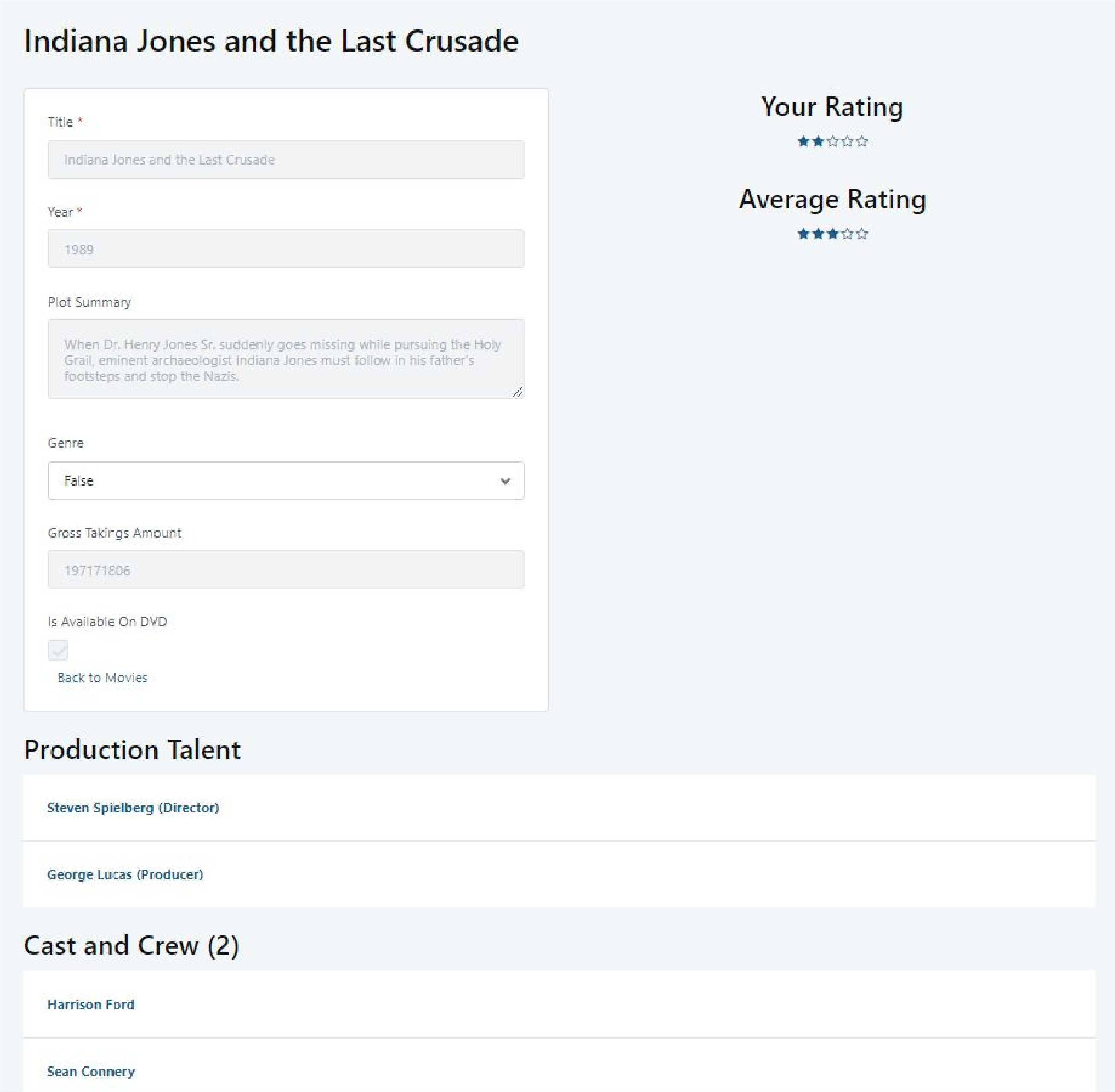
To accomplish this we will:

1. Create a Block to display a number of stars.
2. Make sure that the rating is visible to any end-user by the number of filled stars.
3. Display the average rating of a movie.
4. Without implementing the stars a second time, display a set of stars for the user to rate a movie.

5. Make the stars “clickable” and trigger the logic to create/update the rating of that user for that particular movie.
6. Guarantee that the average rating stars are not clickable.
7. Change the application to allow only Registered users to rate a movie.

A user can rate the same movie more than once. In those cases, the application should consider this as an update of the previous rating, and not as a new rating from that user.

At the end of the exercise, the MovieDetail Screen should look like the following screenshot:



Indiana Jones and the Last Crusade

Title *
Indiana Jones and the Last Crusade

Year *
1989

Plot Summary
When Dr. Henry Jones Sr. suddenly goes missing while pursuing the Holy Grail, eminent archaeologist Indiana Jones must follow in his father's footsteps and stop the Nazis.

Genre
False

Gross Takings Amount
197171806

Is Available On DVD
☑

Back to Movies

Your Rating
★★☆☆☆

Average Rating
★★★☆☆

Production Talent

Steven Spielberg (Director)

George Lucas (Producer)

Cast and Crew (2)

Harrison Ford

Sean Connery

Resources

For this exercise, we will need to use a Client Action defined in the **StarRating** module. This module is already available in the Boot Camp environment, so it is ready to be used.

Hands-on

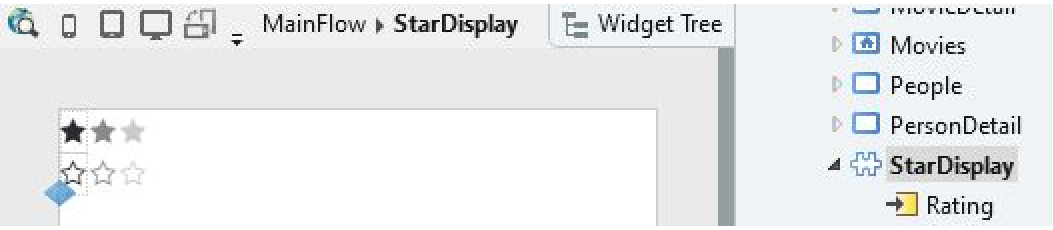
In this exercise we will expand the UI of our application with the functionality that allows users to rate movies. The MovieDetail Screen will allow a user to rate a movie, by clicking on the star respective to its rating.

Also, the Screen will display the average rating of a movie using the stars.

To avoid defining the UI and logic for the stars, we will use Blocks to create some reusable UI that we can apply for both scenarios.

Create the Block to Display the Stars

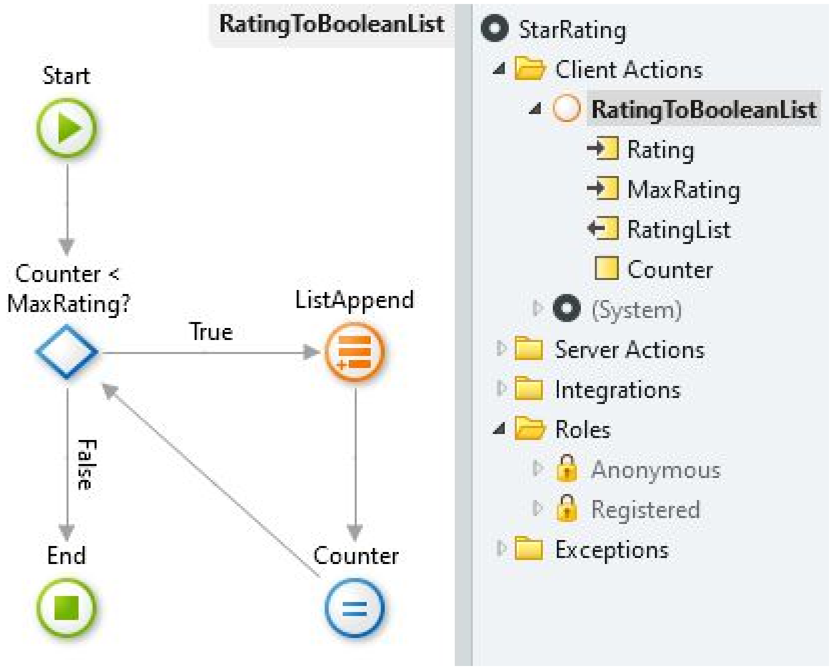
In this first part of the exercise, we will create the StarDisplay Block. This Block will display a certain number of stars, which at this point will be **5**. These stars will appear filled or hollowed depending on the rating given by the user. The rating can be passed to the Block via input parameter. The Block should look something like this in Service Studio:



To create this Block we need to use icons, the **star** icon for the filled and hollowed star (there are two icons with the same name).

From Ratings to Booleans

Then, we need to create a condition that will define when a star appears filled or hollowed. The StarRating module installed in the environment already has an Action, called **RatingToBooleanList**, that expects a Rating and a Max Rating and creates a List of Booleans.



Let’s see some examples of what this Action does:

- With Rating = 3 and Max Rating = 5, the list would be: {True, True, True, False, False}
- With Rating = 1 and Max Rating = 4, the list would be: {True, False, False, False}
- WIth Rating = 3 and Max Rating = 3, the list would be: {True, True, True}

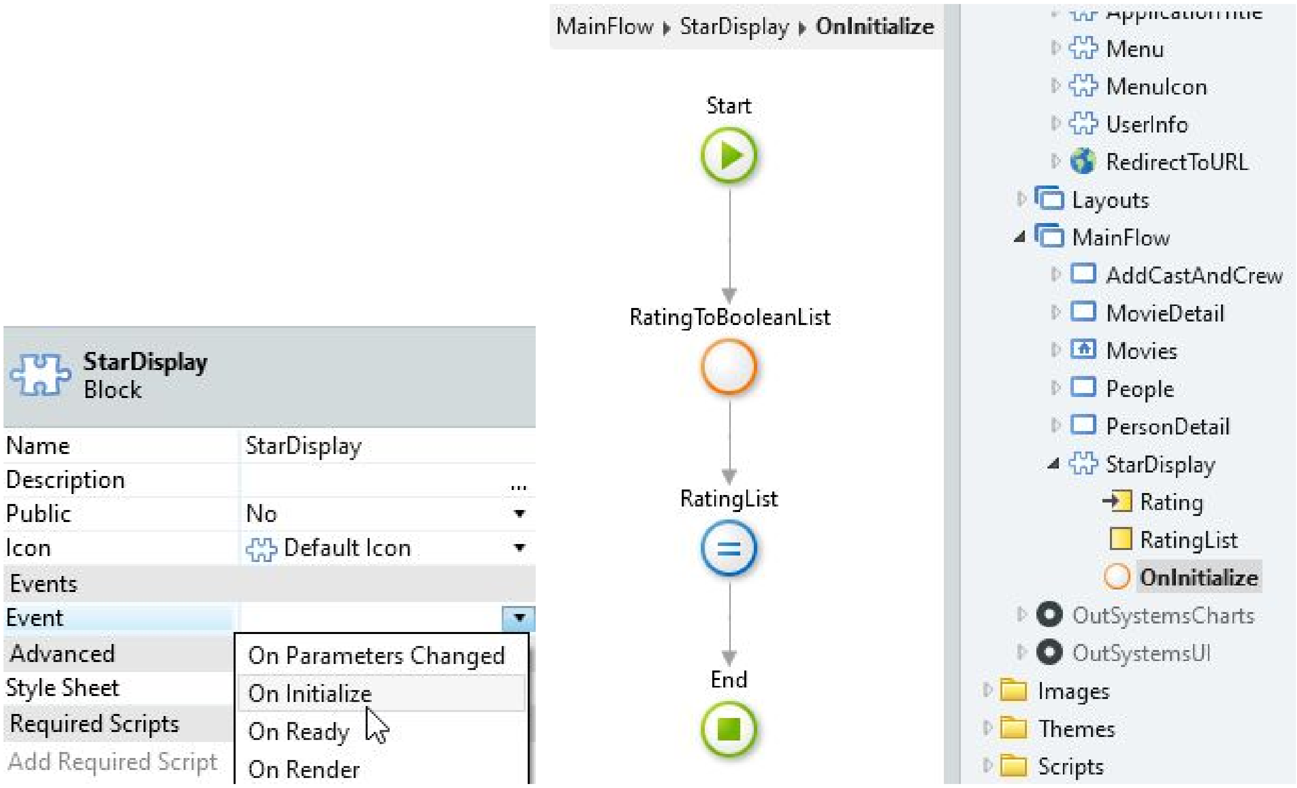
If we consider our scenario with the stars, the output of the Action can be used as a source of a List widget, with the icons inside. Then, we can leverage the boolean values to determine if the stars should appear filled or hollowed.

Now, how can we use the output of the Action in our Block?

From Booleans to Stars

To store the List of Booleans in the scope of the Block we need a Local Variable. That Local Variable can then be used as the Source of the List and its values used to define the filled/hollow stars.

However, how do we initialize the List? By default, the Local Variable will be empty. The Blocks have a couple of Events that we will detail later in this course. One in particular, On Initialize, allows executing some logic when the Screen is initializing. We can use that to call the RatingToBooleanList Action and assign its output to our Local Variable.



MainFlow ▸ StarDisplay ▸ **OnInitialize**

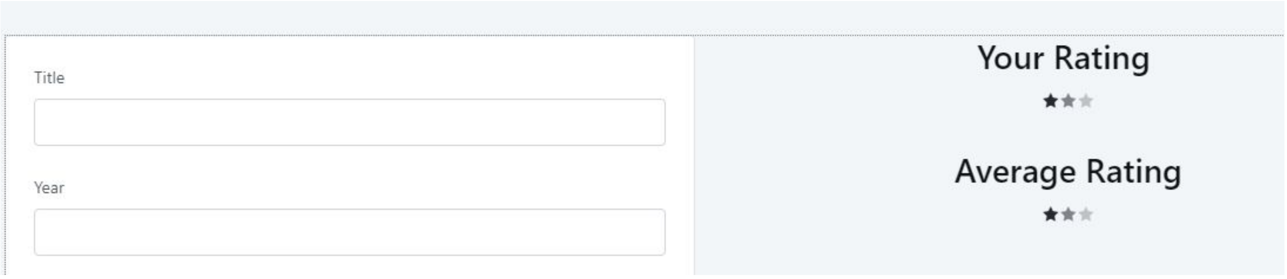
StarDisplay
Block

Name	StarDisplay
Description	...
Public	No
Icon	Default Icon
Events	
Event	
Advanced	
Style Sheet	
Required Scripts	
Add Required Script	

NOTE: We can get more information on the Screen/Blocks events and on the On Initialize Event in specific here.

Add the Blocks to the Screen

On the MovieDetail Screen, we want to display the end-user rating and the average rating, side-by-side with the Form. For that, we need to make the Form to have a width of 6 columns and use the space on the right to display the ratings.



In terms of UI, we can set the *Your Rating* and *Average Rating* texts to the styles that we want to. In the example, both texts use the **heading2** style class.

As we know, the Block has an input parameter for the Rating. In each instance of the Block in the MovieDetail, we need to pass a value for the Rating, so that the Block knows that to display the stars appropriately. To do that, we need some Aggregates:

- One to fetch the rating that the logged in user gave to the movie (if any)
- One to fetch the average rating from all users on that movie.

Make the Stars Clickable

Now, let's make the stars clickable to enable the users to rate their movies. To achieve this, we need to follow the next steps:

- In the Block, create a Link on the stars. When clicked, the Link should trigger a non-mandatory Event and pass the clicked star as input.

  Hint: The Lists have a property called CurrentRowNumber that can be useful :)

- Create a Handler for the instance of the Block in the MovieDetail Screen that represents the user rating. This handler needs to:
  - Create/update the rating of the movie in the database.
  - Re-execute the Aggregate that fetches the Average Rating.

- Guarantee that the Block will display the correct filled stars. Since we're updating the ratings, the input parameter of the Block will change (in one case with the new rating and in the other with the new average), so the stars will also change.

  Hint: Just like we did in the On Initialize, we need to use the RatingToBooleanList Action. Is there any other Block Event we can use?!

Make the Average Rating Non-Clickable

We're almost done with the MovieDetail Screen, except for two details. The first one is related to the star clicking.

While it makes sense to have users rate a movie by clicking on the stars under Your Rating, in the average rating section that does not make sense.

Having the same Block with two different behaviors, will make us change our logic a bit and add a new parameter to the Block, to allow or not the users to click on the stars. We can then leverage this to control inside the Block when the star is clickable.

Make the Rating a Feature for Registered Users

FInally, we only want registered users to rate movies. An Anonymous user will not be able to click on the stars. The Security exercise can be used as an inspiration to implement this behavior.