

Master-Detail

Exercise

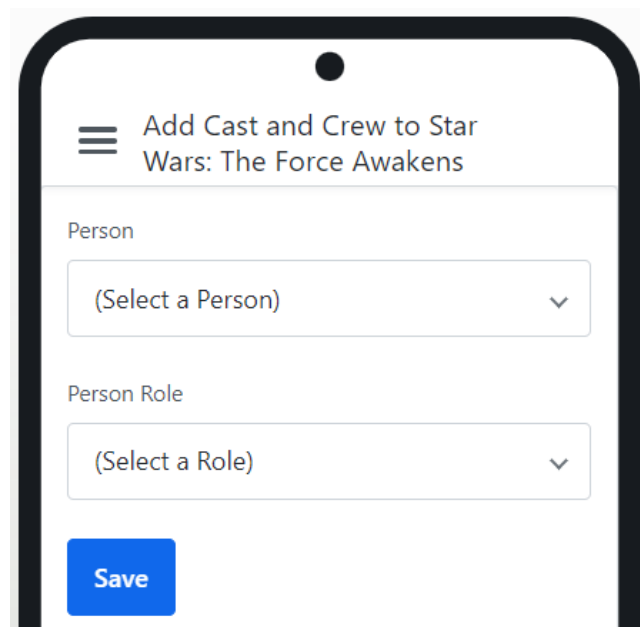
Outline	2
Hands-on	5
Add the Genre Field to the Movie Form	5
Add Cast and Crew to a Movie	5
Build the Screen	5
Styles	6
Creating the new Record in the Database	8
Navigate to the AddCastAndCrew Screen	8
List the Production Talent and Cast/Crew	9
Production Talent	9
Cast and Crew	10

Outline

In this exercise, we will extend our UI functionality. First, we will add a feature to add cast/crew to a movie. For that, we will require:

- A new Screen where we can add a person with a certain role to a movie.
- A link on the MovieDetail Screen to navigate to the new Screen that allows adding a cast and crew to that movie.
- This link should be the only way to get to that Screen when navigating the app.

The new Screen should look like this:




The mockup shows a mobile app screen titled "Add Cast and Crew to Star Wars: The Force Awakens". It features two dropdown menus: "Person" with the placeholder text "(Select a Person)" and "Person Role" with the placeholder text "(Select a Role)". Below these is a blue "Save" button.

Then, in the second part of the exercise, we want to display the following information on the MovieDetail Screen:

- List of the Production Talent involved in the movie (Directors and Producers).
- List of the cast and crew of the movie (Actors and Crew)

Also, we want to add a new field to the MovieDetail Form to allow a user to select a genre for the movie.

The MovieDetail Screen should look like this in the end:

 Star Wars: The Force Awakens

Title *

Star Wars: The Force Awakens

Year *

2015

Plot Summary

Three decades after the defeat of the Gal

Genre

(Select a Genre) ▼

Gross Takings

815843529

Is Available On DVD

☒ [Back to Movies](#)

Production Talent

[Steven Spielberg \(Director\)](#)

[George Lucas \(Producer\)](#)

Cast and Crew

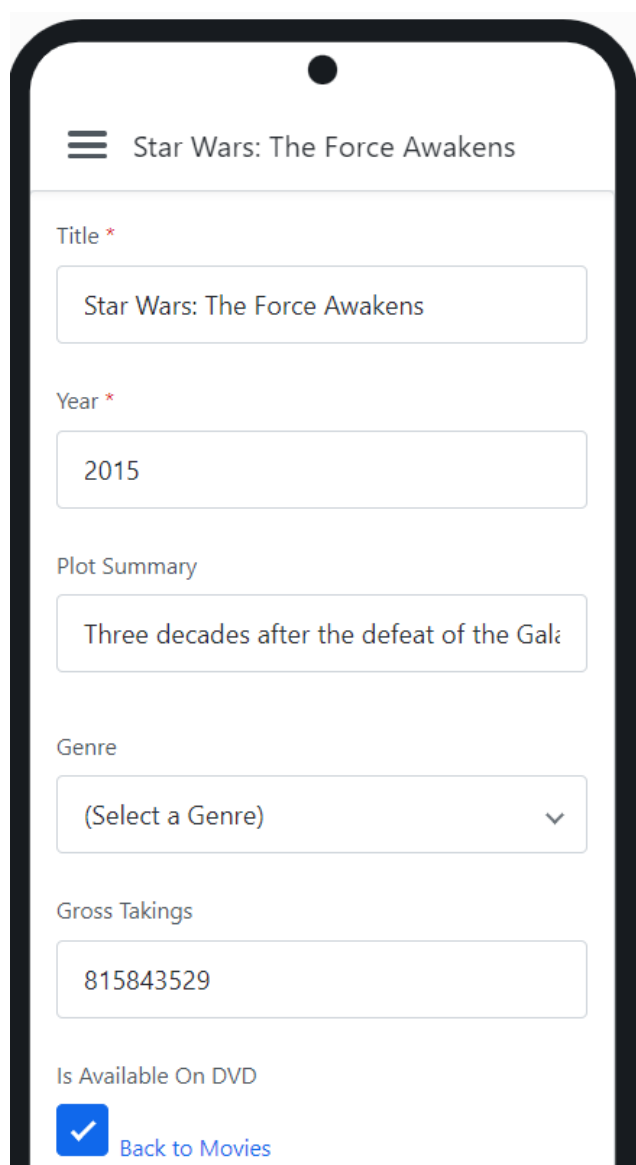
[Harrison Ford \(Actor\)](#)

Hands-on

In this exercise, we will extend the UI of our app to allow an end-user to add cast and crew members to a movie. Later, we want that information displayed next to the movie details on the MovieDetail Screen. However, let's start by adding the movie genre field to the Form.

Add the Genre Field to the Movie Form

In the last exercise, we added the GenreId attribute to the Movie Entity. So now, it's time to allow the end-user to set a genre to a movie in the MovieDetail Screen. Just place the Genre role in a dropdown right below the plot summary.



Star Wars: The Force Awakens

Title *

Star Wars: The Force Awakens

Year *

2015

Plot Summary

Three decades after the defeat of the Gal

Genre

(Select a Genre) ▼

Gross Takings

815843529

Is Available On DVD

☒ Back to Movies

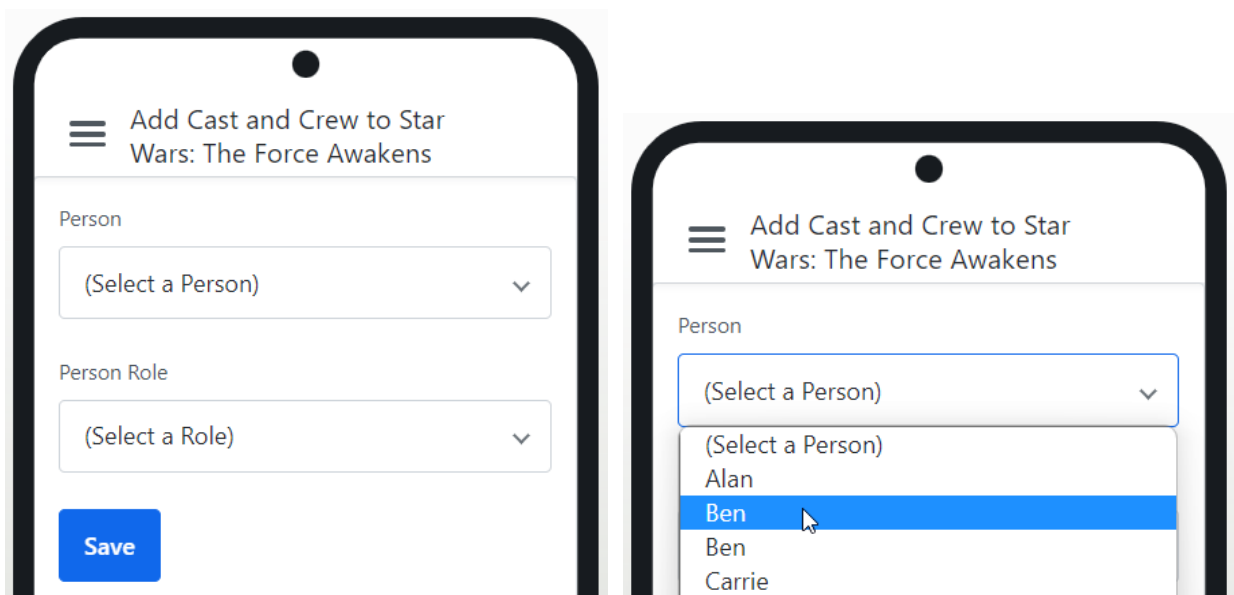
Add Cast and Crew to a Movie

At this point, it's crucial to understand where the end-user will be able to add a member of the cast/crew to a movie. That will take place on a new Screen, the *AddCastAndCrew*.

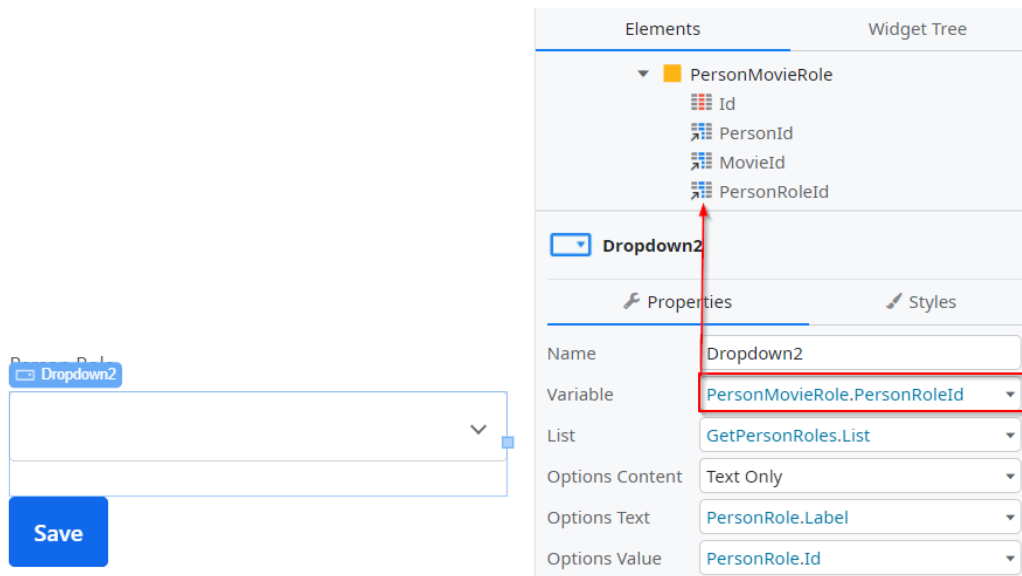
Build the Screen

With that in mind, let's create a new **Empty Screen**, call it *AddCastAndCrew*, and make it accessible by **Everyone**. Since we're coming from the *MovieDetail* Screen, where we already know the *MovieId*, we can leverage that by passing that information to this new Screen via Input Parameter.

The objective of this Screen is to add a person with a certain role to a movie. Translating that to our data model, we want to add a new record to the *PersonMovieRole* Entity. Since we already have the identifier of the movie, we just need the end-user to select the person and the role. The Screen should look like the following screenshots:

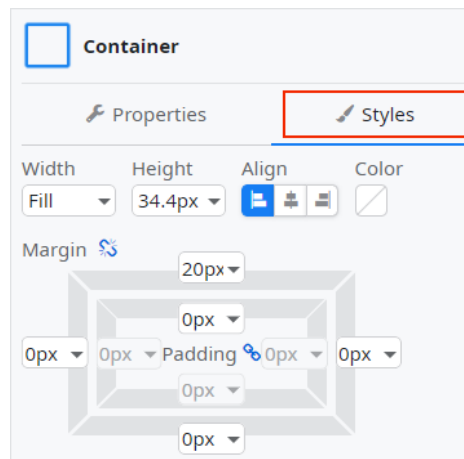


Hint: Since we're working with a scenario of creating new records for the *PersonMovieRole*, with no updates, it is not relevant to fetch data from the *PersonMovieRole* Entity. So, we can use a **Local Variable** of *PersonMovieRole* data type to help us save the info (person and role) that the user selected.

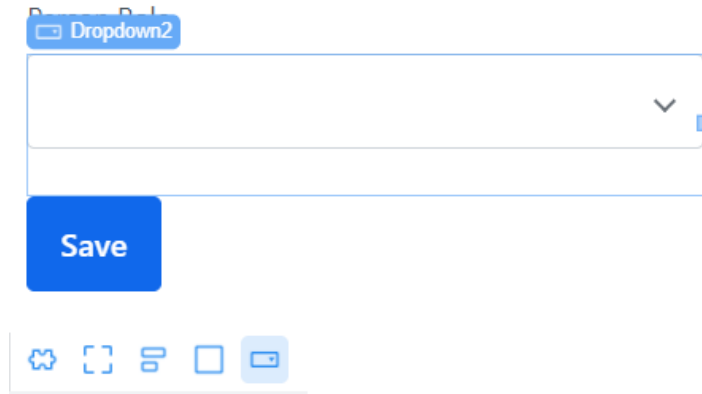


Styles

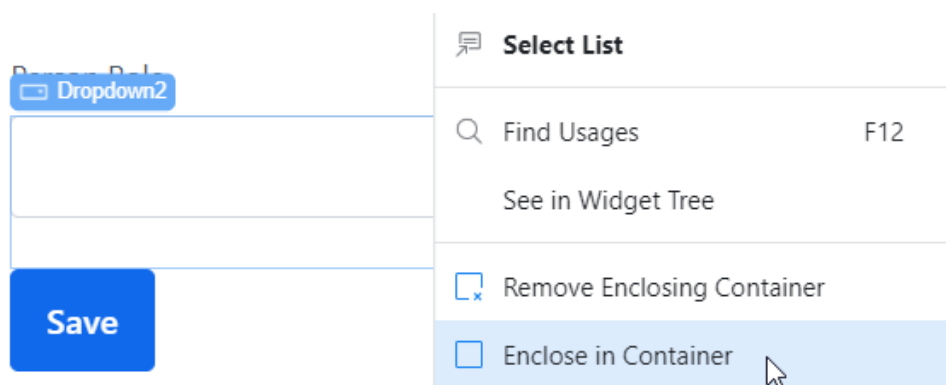
To help separate the widgets on the Screen, or to add margins between them, we can use Containers. These properties are defined in the Styles area of a Container, where we can define margins, change the size and color of the content inside, set the width, etc.



Some of the widgets already have some Containers enclosing them. We can see that when we click on the widget and look at the widget's breadcrumb at the bottom of the Screen.



If the widgets do not have any Container enclosing them, it is easy to add that by double-clicking on it and selecting the correct option.



Challenge: Can we make the person's full name appear in the dropdown instead of just the first name? :)

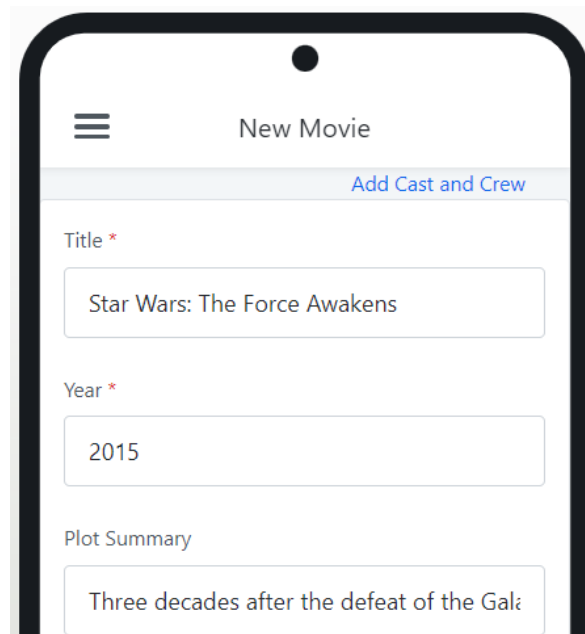
Creating the new Record in the Database

So, now that we're done with the UI, we need to handle the logic to save the PersonMovieRole record in the database. The Save Button will trigger that logic, which will be quite similar to what we have done for the movies and people, with the slight difference that here we only need to create records (not update them!).

Hint: The Form only has an input field for the person and the role. However, the PersonMovieRole Entity has three reference attributes: PersonId, MovieId, and PersonRoleId. So, we need to explicitly assign the identifier of the movie, which is already known, to the PersonMovieRole MovieId attribute. This is very common when the Form does not have fields for all the Entity's attributes and we need to programmatically assign proper values to a variable or attribute before adding the record to the database. This helps avoid errors in the database

Navigate to the AddCastAndCrew Screen

Just like in a previous exercise, we need to have a way to navigate to this new Screen. To solve that, we need to go back to the MovieDetail Screen and create a link to this new Screen.



List the Production Talent and Cast/Crew

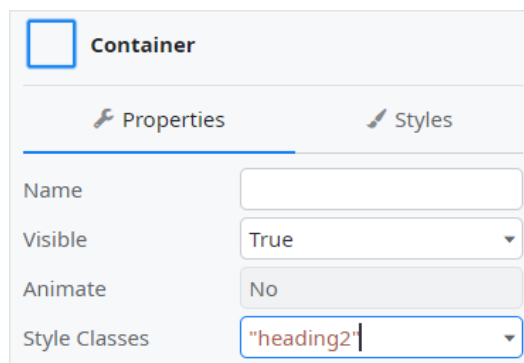
Production Talent

Now that we can add cast/crew members to a movie, we can display that information on the MovieDetail Screen.

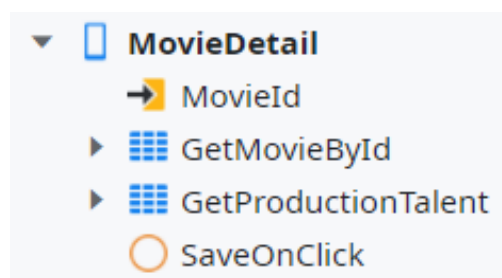
We want to list the directors and producers that worked in the movie below the Form, using a List with the information appearing as **Name (Role)**.

Production Talent	
Steven Spielberg	(Director)
George Lucas	(Producer)

To get the Production Talent section of the Screen to look like this, we need to do some UI tweaks. Below the Form, we can add the static text Production Talent, enclose it in a **Container** and then apply the “*heading2*” **Style Class**. For margins, we should switch to the **Styles Editor** of the Container and add *20 px* to the upper margin and *10 px* to the lower margin.



For the List, we need to fetch all the People whose Role is Director or Producer and who participated in this movie. We can call that Aggregate, GetProductionTalent.



Before we finish, make sure that every member of the production talent has a link to its PersonDetail Screen.

Cast and Crew

Now, for the Cast and Crew, let's follow the same strategy and add a new List to display the actors and crew members involved in the movie.

Production Talent

[Steven Spielberg \(Director\)](#)

[George Lucas \(Producer\)](#)

Cast and Crew

[Harrison Ford \(Actor\)](#)