

Modeling Data

Exercise

Outline	1
Resources	1
Hands-on	1
Populating the Entities with Data	3

Outline

In this exercise lab, we will create the data model for the OSMDb app created in the previous exercise. At this stage, the data model will consist of two Entities, **Movie** and **Person**, and two Static Entities, **MovieGenre** and **PersonRole**.

These Entities will represent the movies (Movie) in the database and their genres (MovieGenre), as well as the cast and crew (People) and the role they play in the movies (PersonRole).

The movie genres we will use are Comedy, Action, Drama, and Horror, while the cast and crew roles used will be Director, Producer, Actor, and Crew.

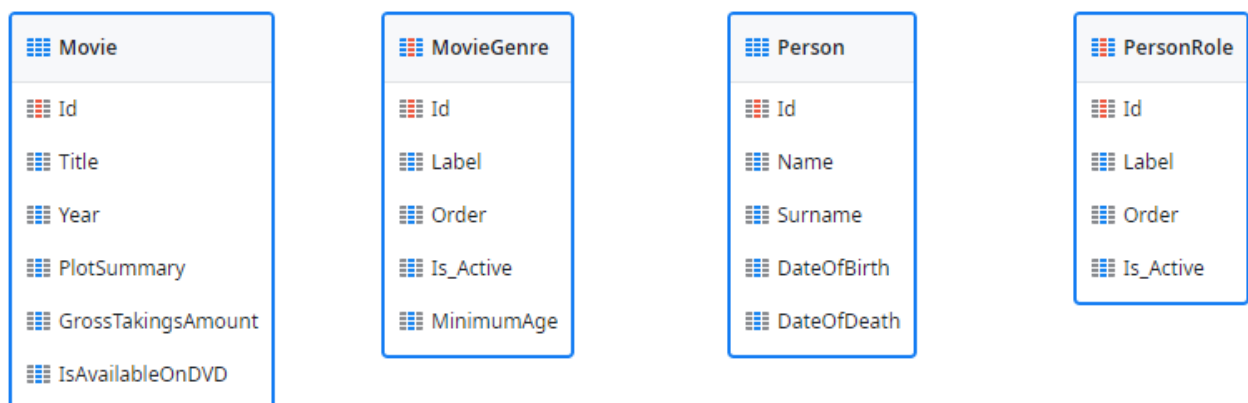
Resources

For this exercise, we will use two Excel files: *Movies.xlsx* and *People.xlsx*. These files can be found in the **Resources** folder of the Boot Camp materials.

Hands-on

In this exercise, we will create the data model of the OSMDb app. This data model should consist of two Entities, **Movie** and **Person**, and two Static Entities, **MovieGenre** and **PersonRole**.

The data model should look like the following screenshot:



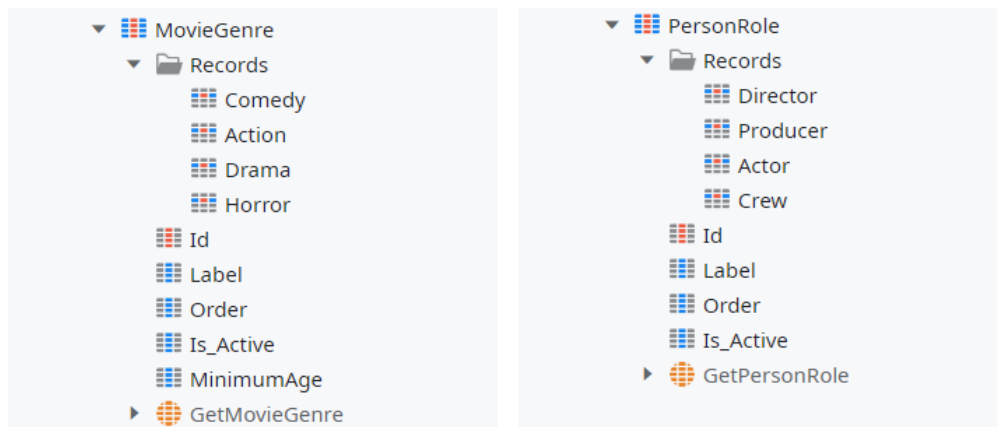
In the *Movie* Entity, the **Title** and **Year** should be mandatory in the Entity's properties. Also, make sure to set a proper **Length** to the **PlotSummary** attribute since it's a longer text. The attributes **Name**, **Surname**, and **DateOfBirth** of the *Person* Entity should be mandatory.

The *MovieGenre* Static Entity should have an extra attribute regarding the minimum age for a person to watch a movie of that particular genre. The genres that we will have in the database, plus the respective minimum age are:

- Comedy: 6
- Action: 12
- Drama: 16
- Horror: 18

The *PersonRole* Static Entity should have the following records:

- Director
- Producer
- Actor
- Crew



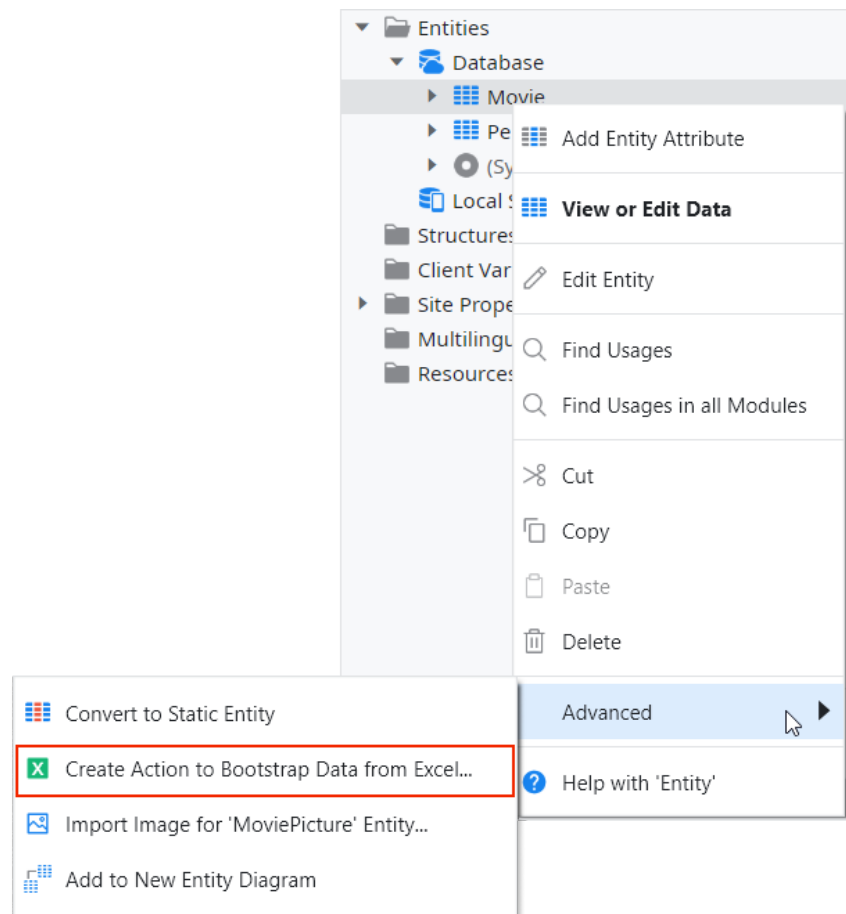
Then, it is important to adjust some other properties, such as the **plural label** of the Person Entity to *People*.

The image shows a screenshot of the 'Person' entity configuration dialog in OutSystems. The dialog has a table at the top with columns 'Id', 'Name', 'Surname', and 'DateOfBirth'. Below the table is a 'More Options' section. In this section, the 'Label' is set to 'Person' and the 'Label (plural)' is set to 'People'. The 'Label (plural)' field is highlighted with a red rectangle. Other options include 'Update Behavior' (set to 'Changed Attributes'), 'Expose Process Events' (unchecked), 'Identifier Attribute' (set to 'Id'), 'Order By Attribute' (empty), 'Label Attribute' (empty), and 'Is Active Attribute' (empty). At the bottom right, there is a 'Close' button.

Populating the Entities with Data

The Entities created are empty. So, we need to populate them with data. On the other hand, the Static Entities already have the data in the database since we manually added the corresponding records for MovieGenre and PersonRole.

For the Movie and Person Entities, use the Excel files in the resources folder to bootstrap data from the files to the Entities. To bootstrap the data, we need to select the respective Entity, in this example, the Movies.



Before we proceed with the bootstrap, we need to ensure that both columns are properly filled out with the names of the columns / attributes. Otherwise, the bootstrap will ignore the columns that do not match and the respective Entity attributes will not be populated with any data. This matching is done per name and by type, meaning that the column name in the Excel file must have the exact same name as the Entity attribute, and the content of the column must match the Entity attribute's data type.

If everything looks ok we can click on Proceed.

Create Action to Bootstrap Data from ...

The 'BootstrapMovies' Action will be created to bootstrap data from Excel Sheet 'Movie'.

Excel Columns	'Movie' Attributes
Title	Title
Year	Year
PlotSummary	PlotSummary
GrossTakingsAmount	GrossTakingsAmount
IsAvailableOnDVD	IsAvailableOnDVD

?

Proceed

Cancel

Note: The logic for fetching the data from the Excel file and adding it to the database is created in the Action **BootstrapMovies**, under the Logic tab. It checks if any Movies currently exist. If not, it imports the Movies from the Excel spreadsheet and creates a Movie in the database, for each row in the spreadsheet. The Excel file will be saved inside the module in the Resources folder under the Data tab. This Action runs when the module is published.