

# Movie Search

## *Application basics*

The movie search application consist of 3 different screens : The search view, the result list and the detail view. On the search view the user is able to pass in a search term and press the search button. After a moment of loading the user will be taken to the result list. This list consist of all the results found. When clicking on one of the results a detail view will be shown with extended details about the found movie.

## *File structure*

The files of the app are separated by their respective screen. Besides these package folders for the screens there are folders for networking, util and properties. Each screen folder at least contains a **fragment, viewModel and viewModelFactory**.

The Search folder also contains a class file for handling the search query input like trimming and passing it through to the viewModel.

The Result list folder contains **adapter, diffcallback, listener and viewHolder** class files which are used to handle all actions and data providing of the recyclerview which is used for the result list.

The Properties folder contains the properties used to store the data of the found movies. The MovieProperty contains the most information about a movie . The other properties like SearchResult and SmallMovieProperty are used to store the retrieved search information and convert it later into the MovieProperty. The NetworkProperty contains the Api url and Api key used to make the api calls.

The Network folder contains the files that setup the api calls to the server. All these calls are packed eventually into coroutines to make sure it runs on a back thread.

The Detail folder contains as said the default files of a screen to display the data.

A Util package folder is also included which holds currently only the bindingAdapters file. This file contains a bindingadapter to convert the posterUrl to an image resource for displaying in the application.

## *The views*

The views of the application are stored in the layouts folder. Besides the main activity view, which redirects with navigation to the starting fragment, the layout folder contains for each screen a fragment view. The last file is a representation of a single result as it would be displayed in the result list fragment. Below is a description of each view:

- **Search** : A text, search bar and search button. While loading a loading text will be shown below the search button.
- **Result list**: A list of result item will be shown
- **Result list item**: An image of the poster, a title, year, director and genre of a specific movie
- **Detail**: Title of the movie, image of the poster, Year of release, director, plot and runtime.

## *Navigation*

In the navigation folder of the resources package, a navigation file can be found. This file contains the navigation between the different screens and the types of arguments that will be passed through the different screens. In this case the movies information of the search result and details about a single movie.

### *Other resources*

The remaining resources contain the fonts, drawables and view resources like string, dimensions, styles and colors.

### *App compatibility*

The app should according to the app gradle file run on an Android device with an api level of 16 or higher.