

**College of Information Technologie**

**Department of Information Technologie**

**Academic Year 2017-2018**

**First Semester**

**Course Code ITCS 518**

**Project 1**

# **Documentation**

Table des matières

[Documentation 1](#_Toc500451068)

[Introduction 3](#_Toc500451069)

[Analysis 3](#_Toc500451070)

[User Interface 3](#_Toc500451071)

[Code Analysis 7](#_Toc500451072)

[Activities 7](#_Toc500451073)

[MainActivity 7](#_Toc500451074)

[DetailActivity 8](#_Toc500451075)

[Adapter 9](#_Toc500451076)

[MoviesAdapter 9](#_Toc500451077)

[Api 9](#_Toc500451078)

[Client 9](#_Toc500451079)

[Service 10](#_Toc500451080)

[Model 10](#_Toc500451081)

[Actor 10](#_Toc500451082)

[ActorResponse 11](#_Toc500451083)

[Movie 12](#_Toc500451084)

[MoviesReponse 13](#_Toc500451085)

[UML 14](#_Toc500451086)

# Introduction

For the Project 1 of Mobile development courses it was asked to create an android application named MovieViewer. This project is similar to the IMDB mobile application.

Included:

* The research of top movies
* The research of popular movies
* The research of popular movies by country
* The research by movie name
* The research by actor
* The film detail page

In bonus :

* a custom icon for the application
* a custom color (orange\_dark) for the status bar with white font
* the landscape mode implementation
* a swipeRefreshLayout for the films list
* a nestedScrollView in the film details page

# Analysis

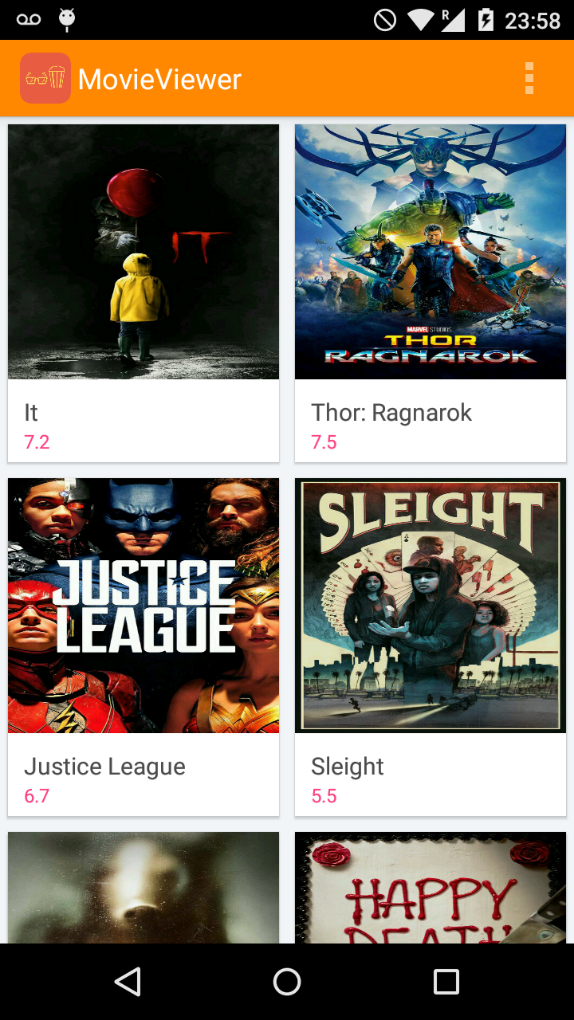
For the realization of this project different types of elements were used:

* SwipeRefreshLayout – For the films list
* NestedScrollView – For the film details page
* The Movie Database API – To have a database of movies and actors
* Retrofit – To create the requests client and convert results into objects
* RecyclerView – For the list of movies
* RecyclerViewAdapter – To convert the requests results into variables
* RadioButton – To select the research option
* EditText – To write the name of the movie or actor in research
* Button – To submit the research.

We divided our programm into 4 parts, the activities, the models, the api and the adapter.

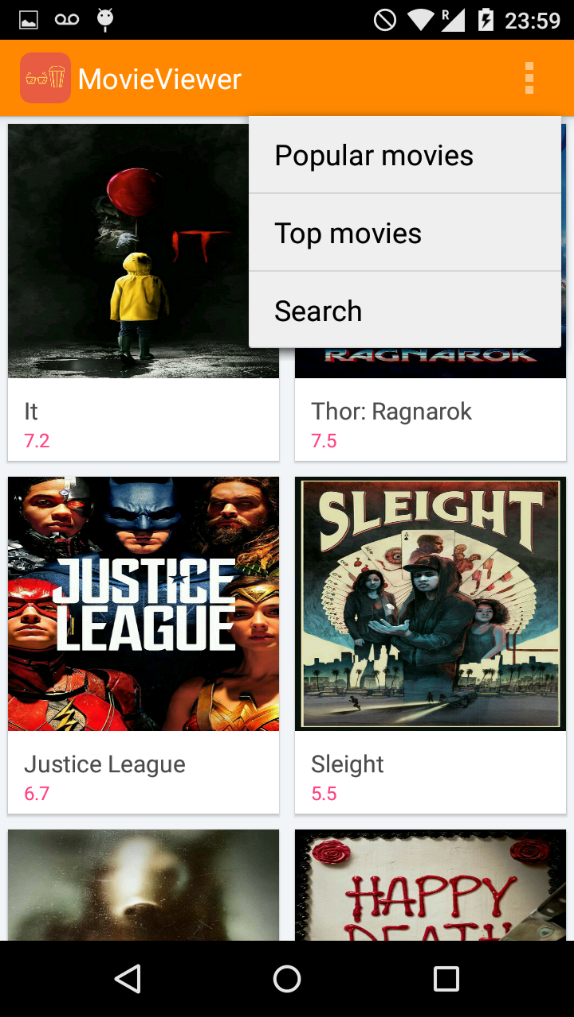
# User Interface

This part describe the user interface of the application:

****

**<- Status Bar with options**

**<- List of movies with poster, name and user rating**

****

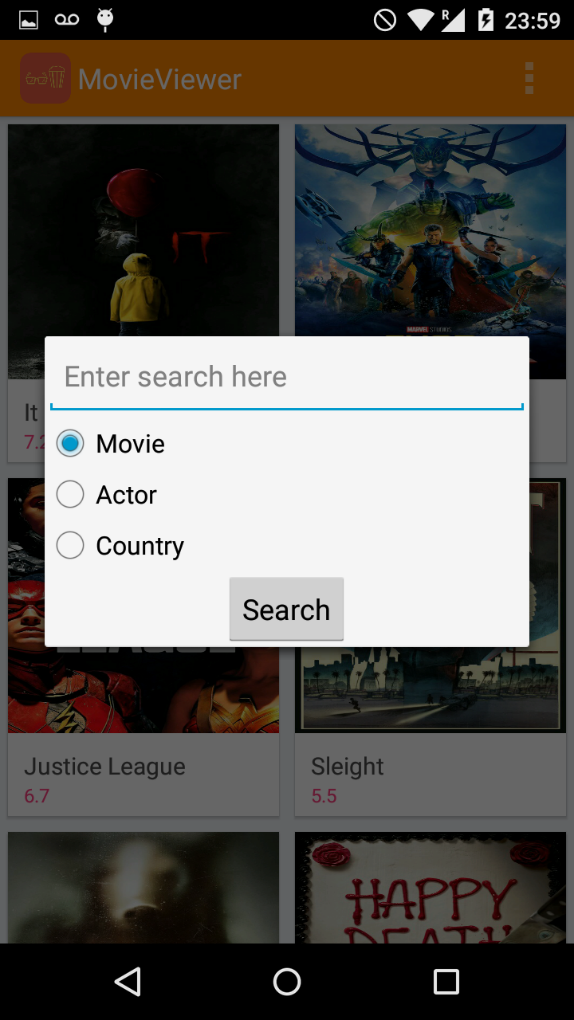
**<- Status Bar with options**

**<- Get the popular movies**

**<- Get the top movies**

**<- Make a search**

**<- List of movies with poster, name and user rating**

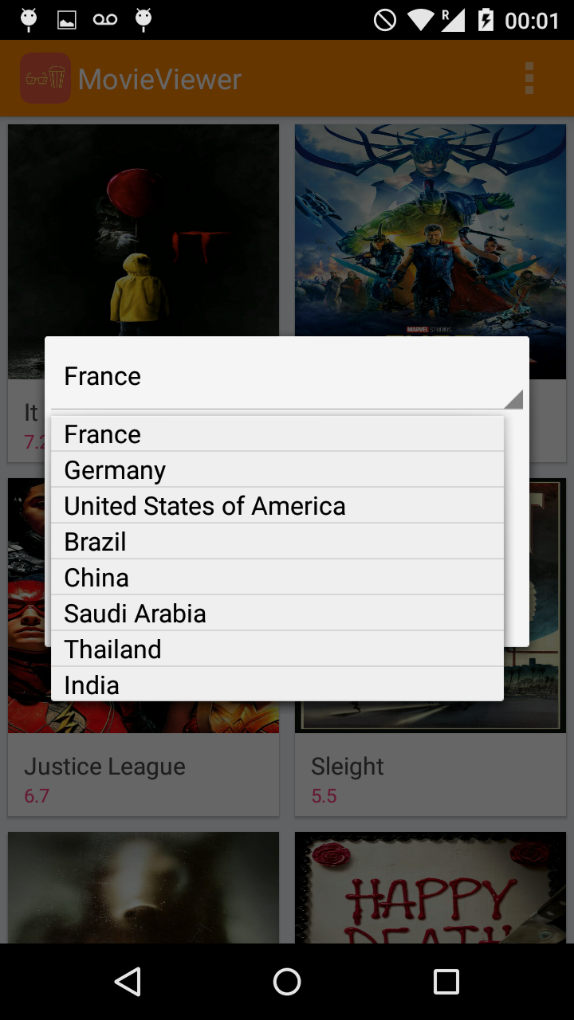
****

**<- EditText to enter the movie or actor name**

**<- RadioButtons to select the search by movie name, actor name or country name (popular movies by country)**

**<- Button to launch the search**

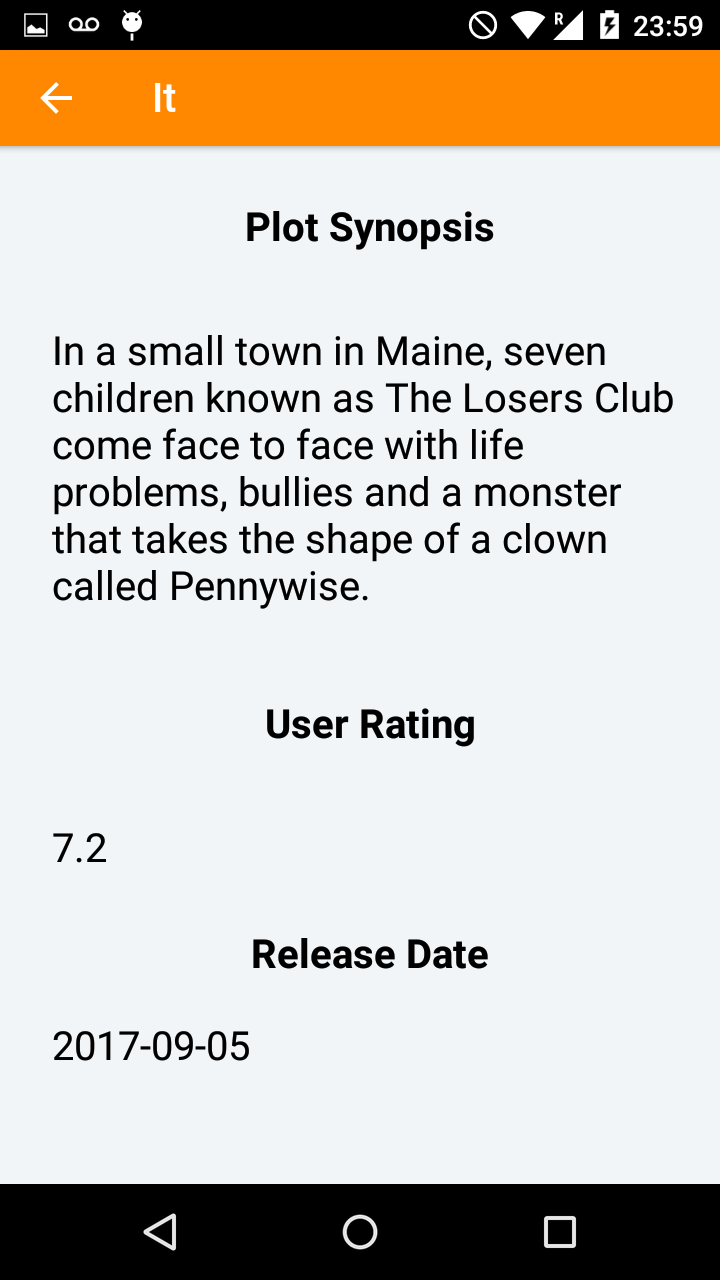
**<- Spinner to select the country**

**<- List of countrys**

**<- Back arrow to the main view**

**<- Poster of the movie**

**<- Name of the movie**

****

**<- Back arrow to the main view and title of the film, if you tap it you return to the movie poster**

**<- Details of the movie**

**<- Status Bar**

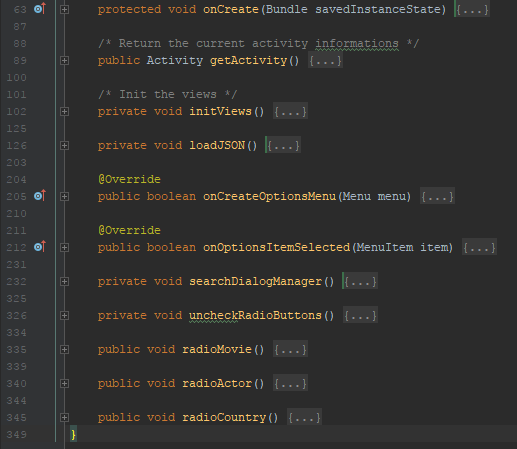
**<- EditText to fill the CPR, first name, last name and email fields**

# **Code Analysis**

The code is divided into 4 parts with 3 packages.

## Activities

### MainActivity



MainActivity is the main view of the activity with the list of movies. Here is the description of each functions:

1. onCreate :

onCreate initialize the principale variables of the MainActivity sucj as the type call (search mode) and the option (name of the actor or movie). It laucnhes the initViews function and initialize the SwipeRefreshLayout and the Spinner ArrayAdapter for the list of country.

1. getActivity :

getActivity return the current Activity.

1. initViews :

initViews initialize the ProgressDialog for the loadging process, the recycler view for the list of movies with its ArrayList and the MoviesAdapter needed. Il also calls the loadJson function.

1. loadJson :

loadJson create the http client and sends the requests in function of it is a search by popularity, top movies, movie name, actor movie or by countrie probability. On the response it refrehs the RecyclerView data.

1. onCreateOptionsMenu :

This function loads the menu\_main.xml file for the custom menu.

1. onoptionsItemSelected :

This function gets the user input to launch the search by popular movie, top movie or the search menu.

1. searchDialogManager :

This function handles the search menu, gets the user input for iether the search by movie name, actor name or country popularity and launches the search.

1. uncheckRadioButtons :

This functions uncheck all the RadioButtons of the search menu.

1. radioMovie :

This function call uncheckRadioButtons and check the movie RadioButton of the search menu.

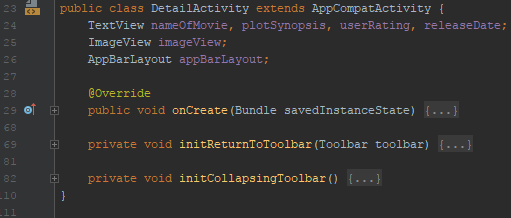
1. radioActor :

This function call uncheckRadioButtons and check the actor RadioButton of the search menu.

1. radioCountry :

This function call uncheckRadioButtons and check the country RadioButton of the search menu.

### DetailActivity



DetailActivity manages the view of movie details. Here is the description of each functions:

1. onCreate :

This function initialize the Toolbar, the SupportActionBar, ImageView for the poster and the EditTexts. Get the data passed by MainActivity and set them for the movie informations. Is also calls initCOllapsingToolbar and initReturnToToolbar.

1. initReturnToToolbar :

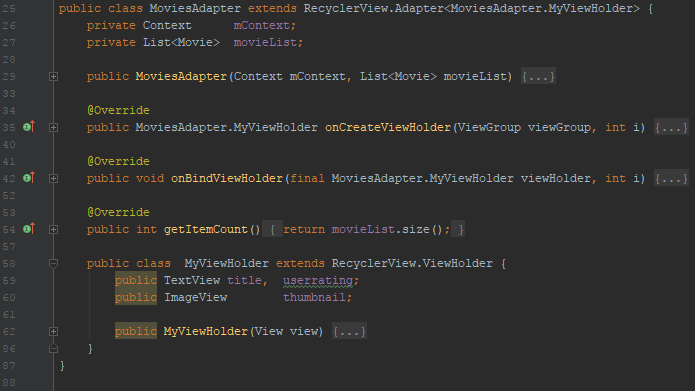
This function is used to return to the poster movie when the user touches the SupportActionBar.

1. initCollapsingToolbar :

This function is used to collapse the SupportActionBar with the movie poster init when the user scrolls down the view.

## Adapter

### MoviesAdapter



MoviesAdapter is used to fill the RecyclerViex film items with the data got by the API. Here is the description of each functions :

1. MoviesAdapter :

Constructor of the class.

1. onCreateViewHolder :

Create and return the ViewHolder.

1. onBindViewHolder :

Fill the data of the movies in the RecyclerView.

1. getItemCount :

Return the size of the List of movies.

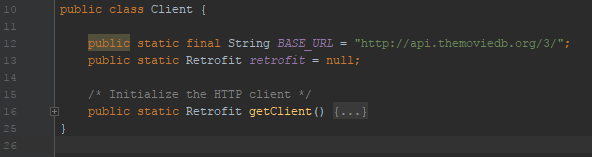
MyViewHolder is the class designed to create the view of the selected movie details :

1. MyViewHolder :

Constructor of the class. Set te details of the movie.

## Api

### Client

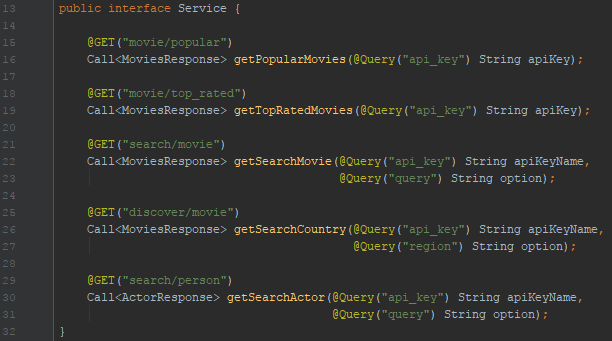


Client is used to create the http client which communicate with The Movie Database Database.

1. getClient :

Create a http client with Retrofit and return it.

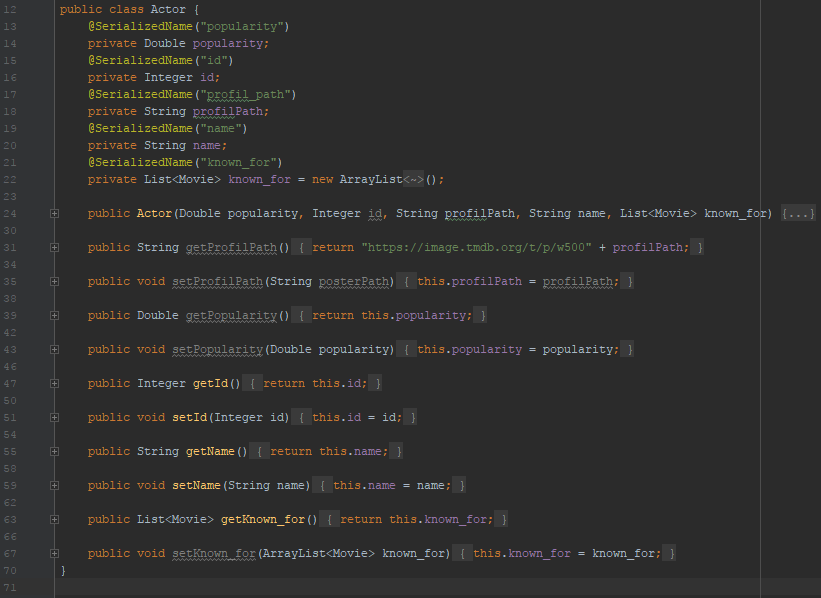
### Service



Service is an interface where the request to the API are defined with Retrofit.

## Model

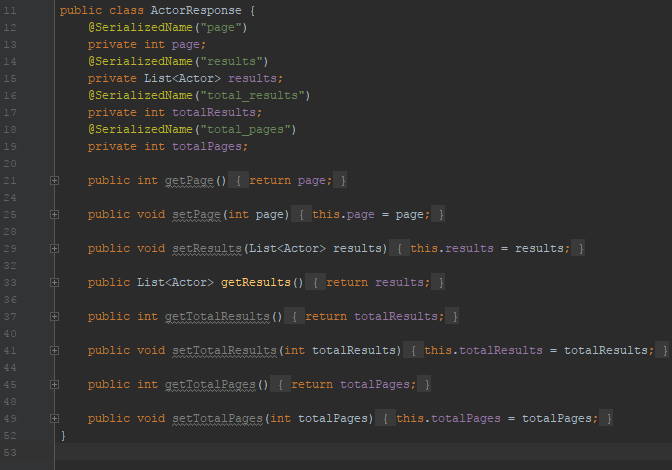
### Actor



Actor is a class designed to create an Actor object with the informations returned by the ActorResponse class.

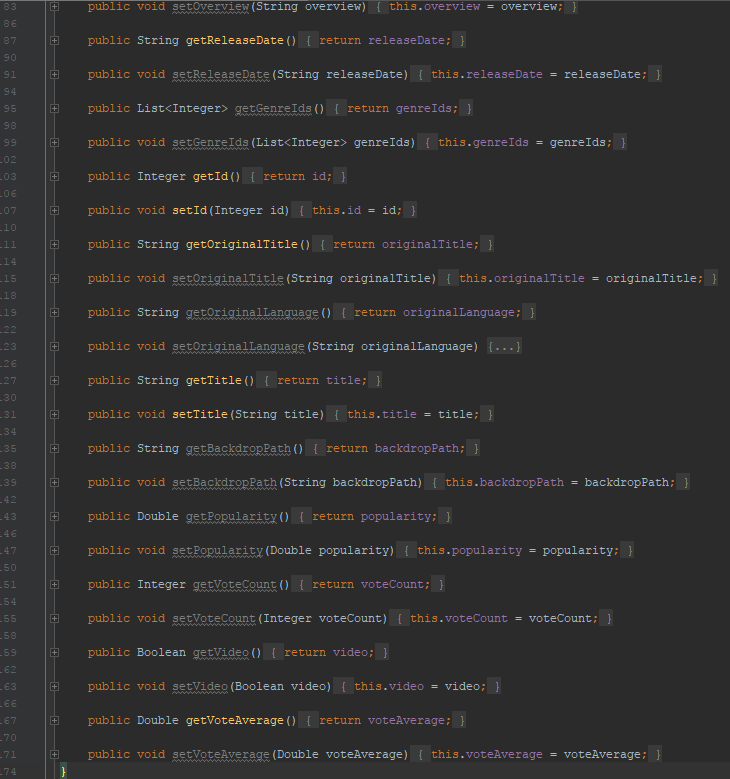
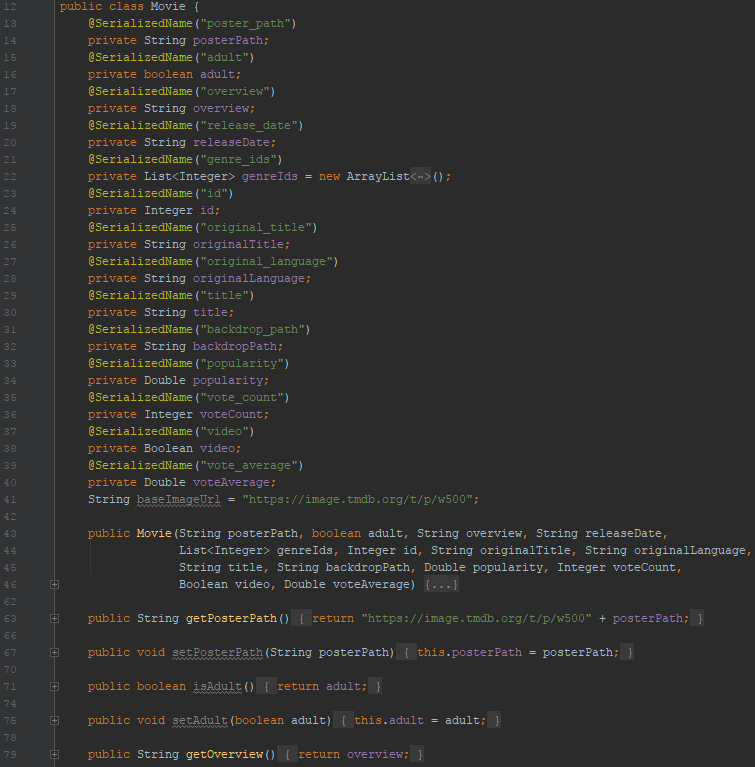
It is composed by a constructor with the concerned informations and the getters and the setters of the object variables.

### ActorResponse



ActorResponse class is designed to return a list of actors get by the http client with Retrofit. It is composed by its getters and setters for its variables and a getResults function which return the List of Actor objects.

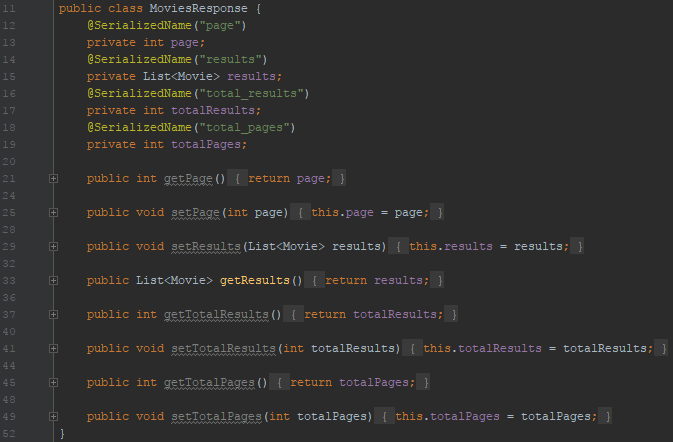
### Movie



Movie class is used to create a Movie object with the data passed in parameter by MovieResponse.

It is composed by its constructor and the getters and setters of his variables.

### MoviesReponse



MoviesReponse is a class designed to create a List of Movie with the datas get by the http client. It is composed by its getters and setters and with a function getResults which returned the list of Movies.

# UML