

Documentation technique

Le projet est disponible sur le Gitlab de l'université :

<https://gitlab.univ-lille.fr/florent.cadet.etu/rendu-android>

ou sur github : <https://github.com/florent-cadet/rendu-android>

Librairies utilisées

- RxJava 2 pour gérer les événements asynchrones.
 - La version 2 a été utilisée car le projet a commencé avant sa dépréciation le 28 février 2021
- Retrofit pour faciliter l'écriture des appels HTTP
- Utilisation de la librairie Gson pour gérer la conversion des objets Java en JSON et inversement
- Stetho pour faciliter le debug des requêtes HTTP
- OkHttp pour gérer les requêtes HTTP envoyés à l'API - The Movie Database (<https://developers.themoviedb.org/3/getting-started>)
- Glide pour le chargement d'images
- Room pour la base de données locale

Choix faits

- Application développée en utilisant l'architecture MVVM qui est l'architecture recommandée par Google, plus adaptée au développement d'applications mobiles avec peu voir pas de dépendances ce qui simplifie les tests.
- Une classe FakeDependencyInjection, qui comme son nom l'indique sert d'injection de dépendances
- L'application est composée de 2 activités. L'activité principale est composée de 2 fragments.

Fonctionnement général

L'application est découpée en 2 parties :

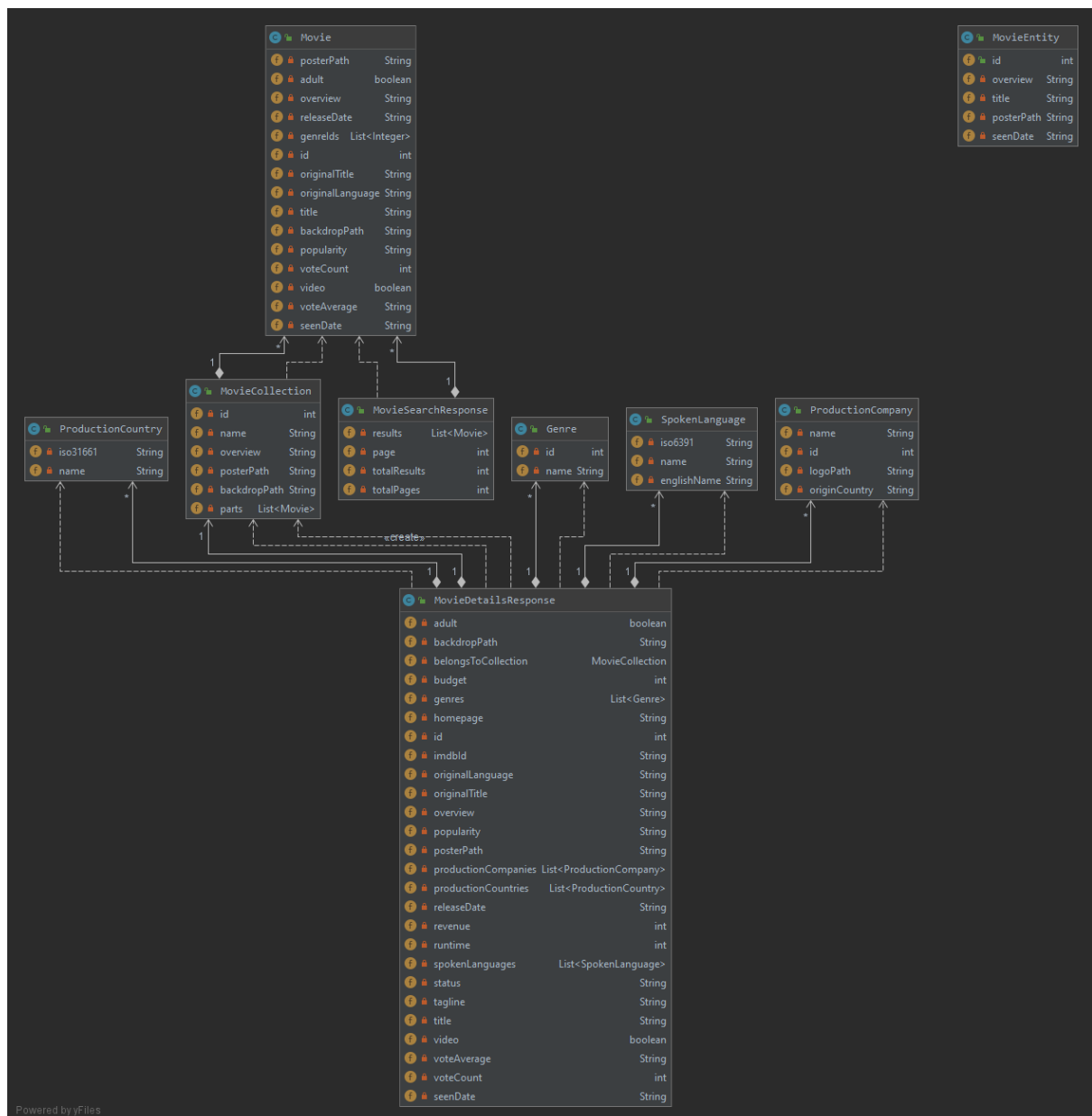
- Une partie presentation possédant les vues et les ViewModel
- Une partie data permettant de gérer la récupération des données depuis l'API ou la BDD locale. Cette partie met en avant un service permettant aux ViewModel de récupérer les informations demandées.

Parc matériel ciblé

La version minimum du sdk est la version 21 correspondant à la version 5.0 d'Android permettant de toucher 94,1% des utilisateurs d'Android.

La version cible du sdk est la version 30 correspondant à la version 11.0 d'Android avec les dernières nouveautés.

Diagramme de classes



Diagrammes de séquences

Diagramme de séquence pour la récupération des données

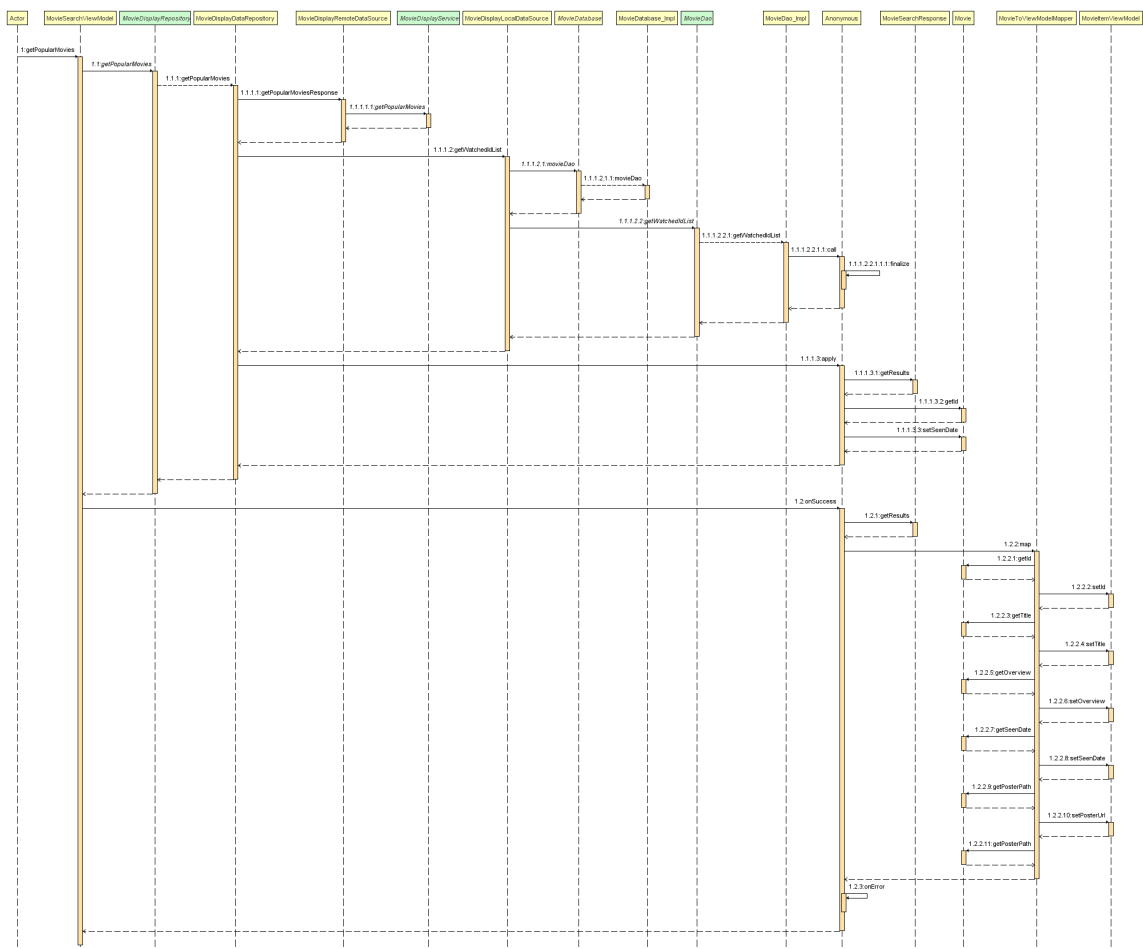
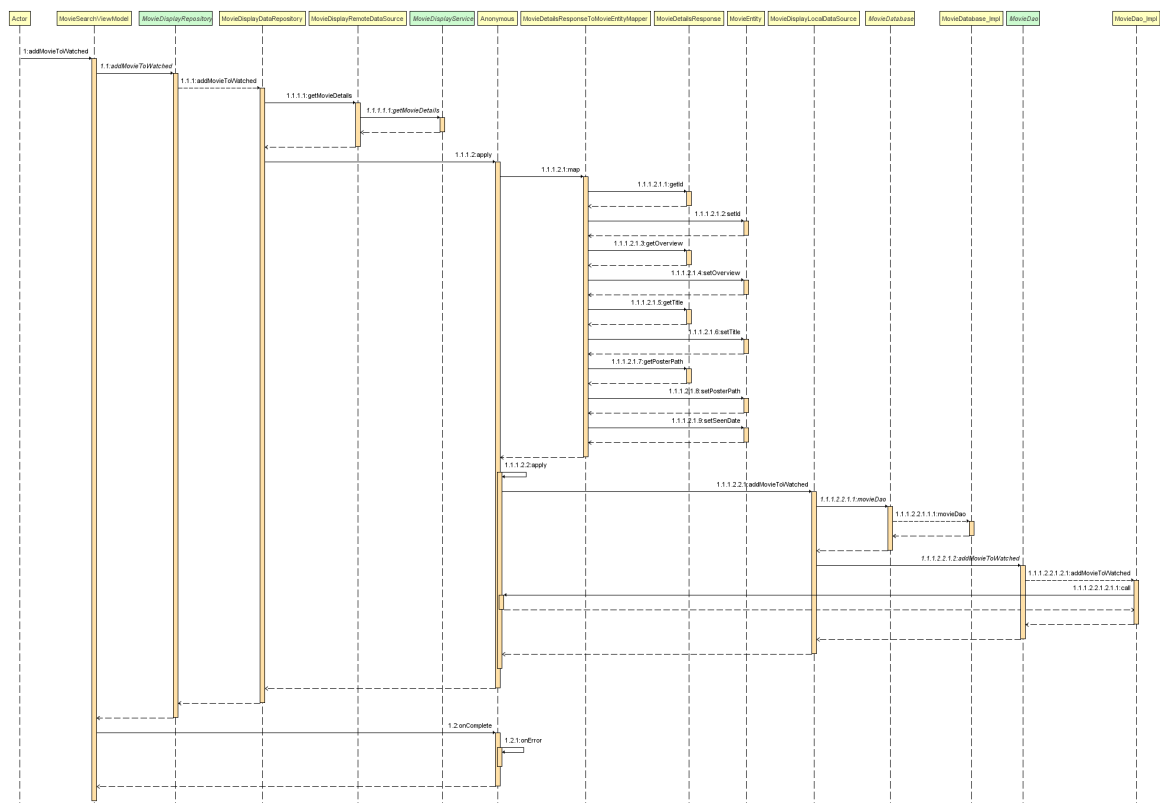


Diagramme de séquence pour la sauvegarde des données



Contrat d'interface

Le contrat d'interface a été réalisé avec swagger. Pour avoir un meilleur visuel, rendez-vous sur <https://editor.swagger.io/> et copier/coller le contenu au format YAML suivant :

swagger: "2.0"

info:

description: API pour le projet Android

version: 1.0.0

title: Projet Android

paths:

/movie/popular:

get:

summary: Get popular movies

produces:

- application/json

parameters:

- name: api_key

in: query

description: The api key required for every request

required: true

type: string

- name: language
in: query
description: Pass a ISO 639-1 value to display translated data for the fields that support it.

required: false
type: string

- name: page
in: query
description: Specify which page to query.
required: false
type: integer

- name: region
in: query
description: Specify a ISO 3166-1 code to filter release dates. Must be uppercase
required: false
type: string

responses:

"200":
description: search results matching criteria
schema:
type: array
items:
\$ref: '#/definitions/MovieSearchResponse'

"401":
description: Invalid API key
schema:
\$ref: '#/definitions/ErrorResponse'

"404":
description: The resource requested could not be found
schema:
\$ref: '#/definitions/ErrorResponse'

/search/movie:

get:

summary: Search a movie

produces:

- application/json

parameters:

- name: api_key
in: query
description: The api key required for every request
required: true
type: string

- name: query
in: query
description: Pass a text query to search. This value should be URI encoded.
required: true
type: string

- name: language

in: query
description: Pass a ISO 639-1 value to display translated data for the fields that support it.

required: false

type: string

- name: page

in: query

description: Specify which page to query.

required: false

type: integer

- name: region

in: query

description: Specify a ISO 3166-1 code to filter release dates. Must be uppercase

required: false

type: string

- name: include_adult

in: query

description: Choose whether to include adult content in the results

required: false

type: boolean

- name: year

in: query

required: false

type: integer

- name: primary_release_year

in: query

required: false

type: integer

responses:

"200":

description: search results matching criteria

schema:

type: array

items:

\$ref: '#/definitions/MovieSearchResponse'

"401":

description: Invalid API key

schema:

\$ref: '#/definitions/ErrorResponse'

"404":

description: The resource requested could not be found

schema:

\$ref: '#/definitions/ErrorResponse'

/movie/{movie_id}:

get:

summary: Get the details of a movie

produces:

- application/json

parameters:

- name: api_key
 - in: query
 - description: The api key required for every request
 - required: true
 - type: string
- name: movie_id
 - in: path
 - description: The id of the movie
 - required: true
 - type: integer
- name: language
 - in: query
 - description: Pass a ISO 639-1 value to display translated data for the fields that support

it.

required: false

type: string

- name: append_to_response

in: query

description: Append requests within the same namespace to the response.

required: false

type: string

responses:

"200":

description: search result matching criteria

schema:

\$ref: '#/definitions/MovieDetailsResponse'

"401":

description: Invalid API key

schema:

\$ref: '#/definitions/ErrorResponse'

"404":

description: The resource requested could not be found

schema:

\$ref: '#/definitions/ErrorResponse'

definitions:

MovieSearchResponse:

properties:

page:

type: integer

totalResults:

type: integer

totalPages:

type: integer

results:

type: array

items:

\$ref: '#/definitions/Movie'

Movie:

properties:

poster_path:

type: string

adult:

type: boolean

overview:

type: string

release_date:

type: string

genre_ids:

type: array

items:

type: integer

id:

type: integer

original_title:

type: string

original_language:

type: string

title:

type: string

backdrop_path:

type: string

popularity:

type: string

vote_count:

type: integer

vote_average:

type: string

ErrorResponse:

properties:

status_message:

type: string

status_code:

type: integer

MovieDetailsResponse:

properties:

adult:

type: boolean

backdrop_path:

type: string

belongs_to_collection:

type: array

items:

\$ref: '#/definitions/MovieCollection'

budget:

type: integer

genres:
 type: array
 items:
 \$ref: '#/definitions/Genre'
homepage:
 type: string
id:
 type: integer
imdb_id:
 type: integer
original_language:
 type: string
original_title:
 type: string
overview:
 type: string
popularity:
 type: string
poster_path:
 type: string
production_companies:
 type: array
 items:
 \$ref: '#/definitions/ProductionCompany'
production_countries:
 type: array
 items:
 \$ref: '#/definitions/ProductionCountry'
release_date:
 type: string
revenue:
 type: integer
runtime:
 type: integer
spoken_languages:
 type: array
 items:
 \$ref: '#/definitions/SpokenLanguage'
status:
 type: string
tagline:
 type: string
title:
 type: string
video:
 type: boolean
vote_count:
 type: integer

```
    vote_average:
      type: string
MovieCollection:
  properties:
    id:
      type: integer
    name:
      type: string
    overview:
      type: string
    poster_path:
      type: string
    backdrop_path:
      type: string
    parts:
      type: array
    items:
      $ref: '#/definitions/Movie'
Genre:
  properties:
    id:
      type: integer
    name:
      type: string
ProductionCompany:
  properties:
    name:
      type: string
    id:
      type: integer
    logo_path:
      type: string
    origin_country:
      type: string
ProductionCountry:
  properties:
    iso_3166_1:
      type: string
    name:
      type: string
SpokenLanguage:
  properties:
    iso_639_1:
      type: string
    name:
      type: string
    english_name:
      type: string
```

