

Audiolly

Dépôt GitHub : <https://github.com/Nouuu/Audiolly>

Application

L'application fonctionne sur Android 5.0 et plus.

Elle a été développée uniquement en Kotlin.

On a utilisé les API de [TheAudioDB](#) pour les rankings, les titres, les albums, les artistes...

Mais aussi l'API [api.lyrics.ovh](#) pour les lyrics. (BONUS)

Fonctionnalité

- Page d'accueil avec onglets
- Classements (charts) des singles et albums
- Fonction de recherche d'un artiste
- Liste des artistes favoris
- Détails d'un artiste
- Fonction de mise en favoris d'un artiste
- Détails d'un album
- Gestion des traductions
- Persistance des données
- Fonction de mise en favoris d'un album
- Liste des favoris : supporter les albums
- Fonction de recherche : supporter la recherche par album

Fonctionnalités bonus

- Écran avec les paroles d'une chanson
- Utilisation du debounce pour le champ de recherche
- Fonction de mise en favoris d'un album
- Les musiques dans les albums redirigent vers la page 'paroles'

- Gestion d'erreurs : dans le cas où une donnée n'arrive pas à se charger, reviens sur la page précédente

Architecture de l'application

```
src/  
├─ api/  
│  └─ response/  
│  └─ {{interactor}}  
├─ features/  
│  └─ {{feature1}}  
│  └─ {{feature2}}  
│  └─ {{feature3}}  
│     └─ {{fragment.kt}}  
├─ models/  
│  └─ {{data class}}  
├─ storage/  
└─ MainActivity.kt
```

API

Dans le package API, on stocke nos **DTO** (data transfer object) dans le dossier **response**, et on a des **interactors** pour communiquer avec les différentes API.

Features

Pour chaque fonctionnalité de l'application, on a utilisé un sous dossier dans le dossier features qui contient les fragments et autre élément de la fonctionnalité en question.

Models

Pour chaque objet "métier" de l'application, on a un model associé qui se trouve dans le dossier model (qui est sous la forme de data class).

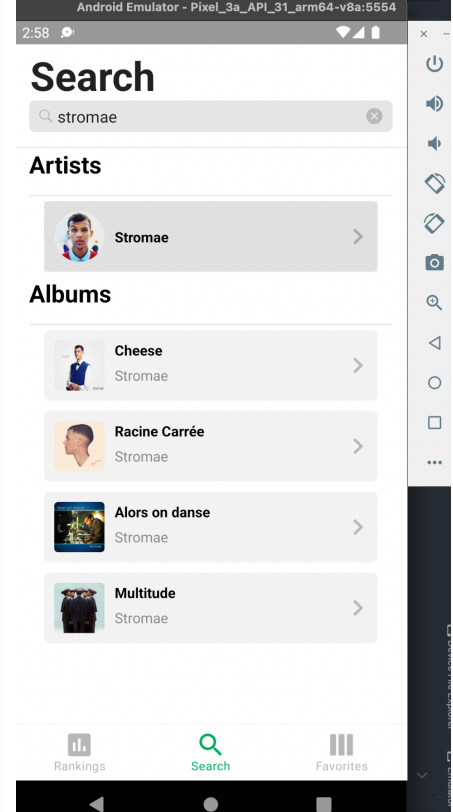
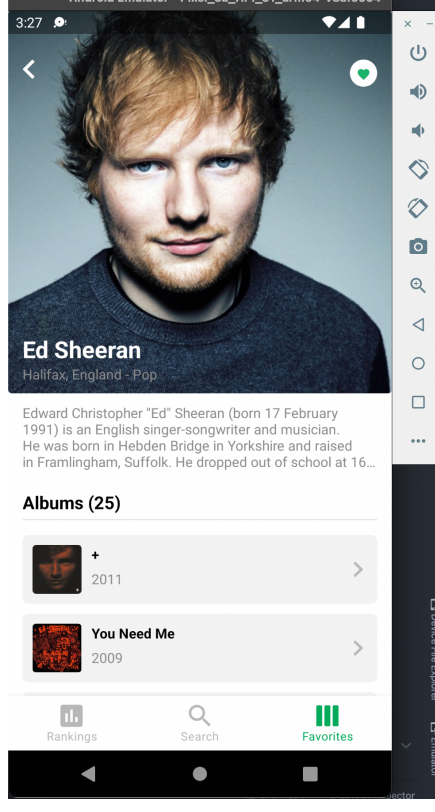
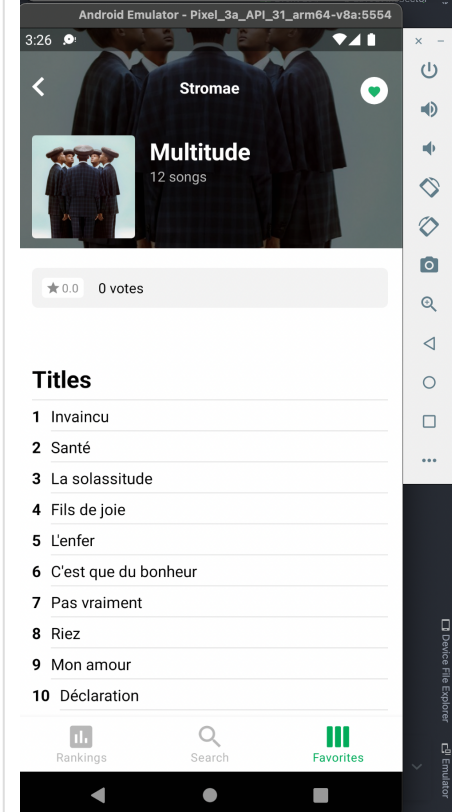
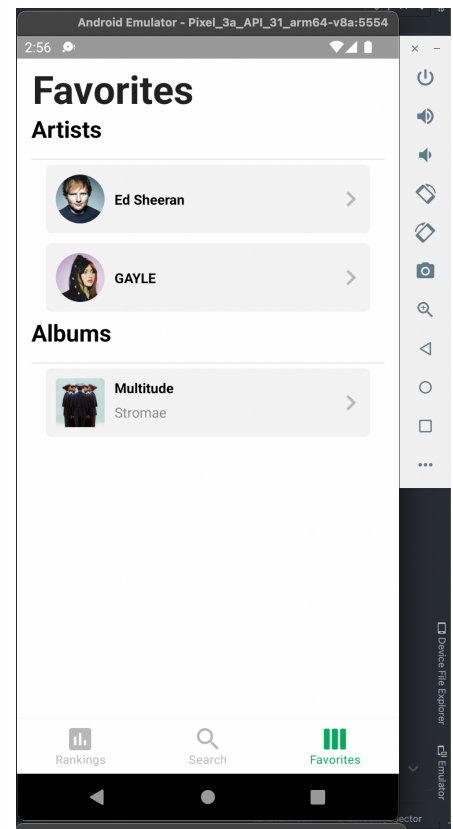
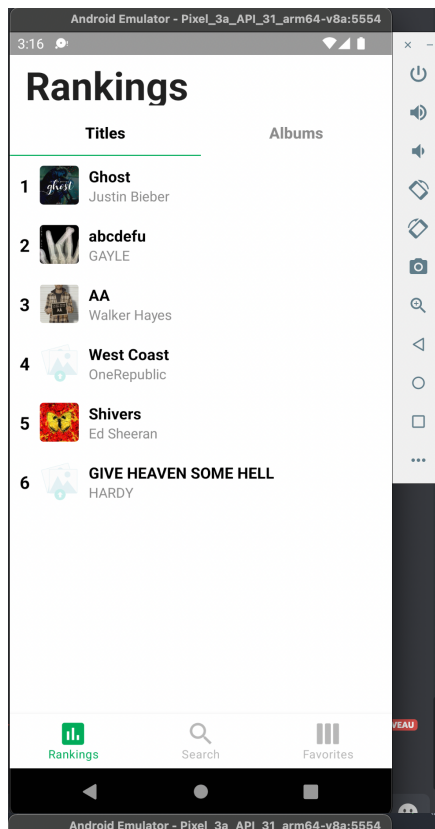
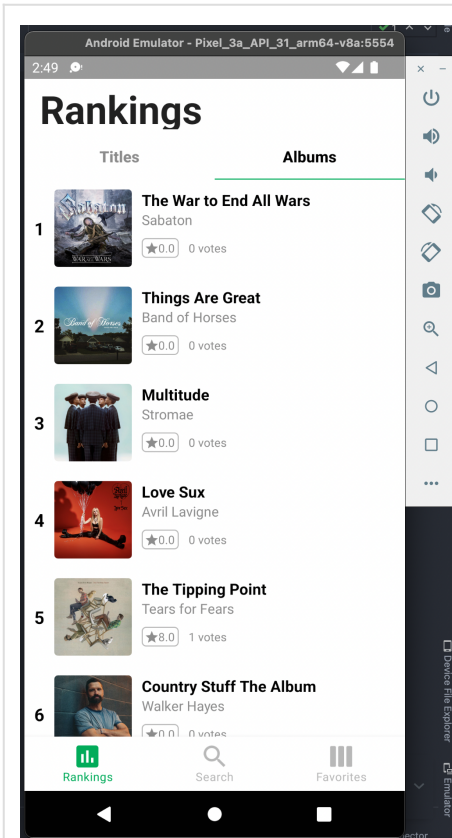
Storage

Dans le dossier storage, on stocke les informations liées à la base de donnée locale SQLite.

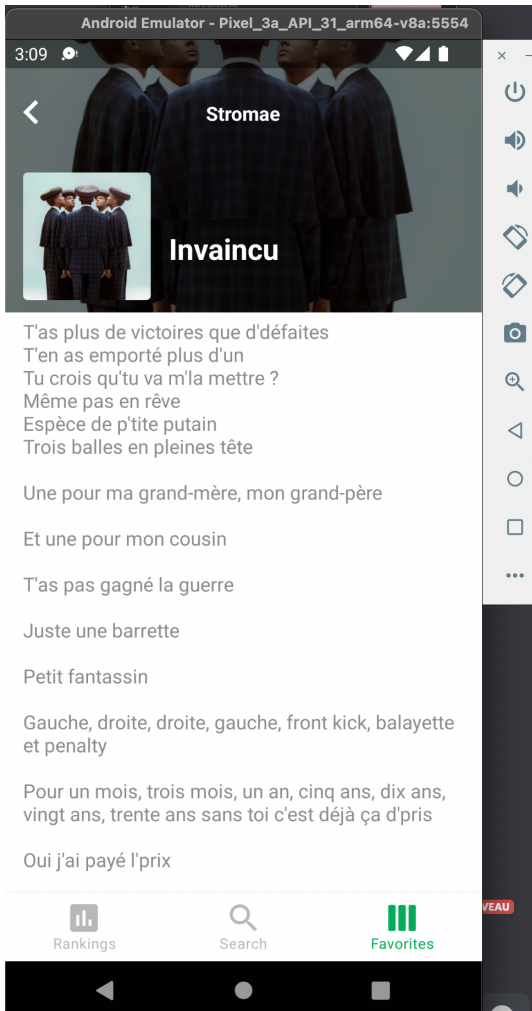
On utilise des DAO (data access objet) pour accéder à nos données comme le veut la librairie **room**.

La persistance des données nous permet de récupérer les informations au chargement.

Quelques images



BONUS



Revision #4

Created 2022-10-02 14:27:04 UTC by Noé Larrieu-Lacoste

Updated 2022-10-02 14:44:05 UTC by Noé Larrieu-Lacoste