Lunatech

IMDB

Documentation

THIRULOGASINGAM Mathusha
30/10/2022

Table des matières

Introduction 2
Décomposition 2
La base de données 2
Le fonctionnel 3
Les technologies 3
L'API 3
L'interface 4
Les technologies 4
Le rôle 4
Axes d'amélioration 4
Kevin Bacon Game 4
NameBasicsConverter 5

Introduction

Ce projet a pour but de créer des entrées utilisateurs afin de :

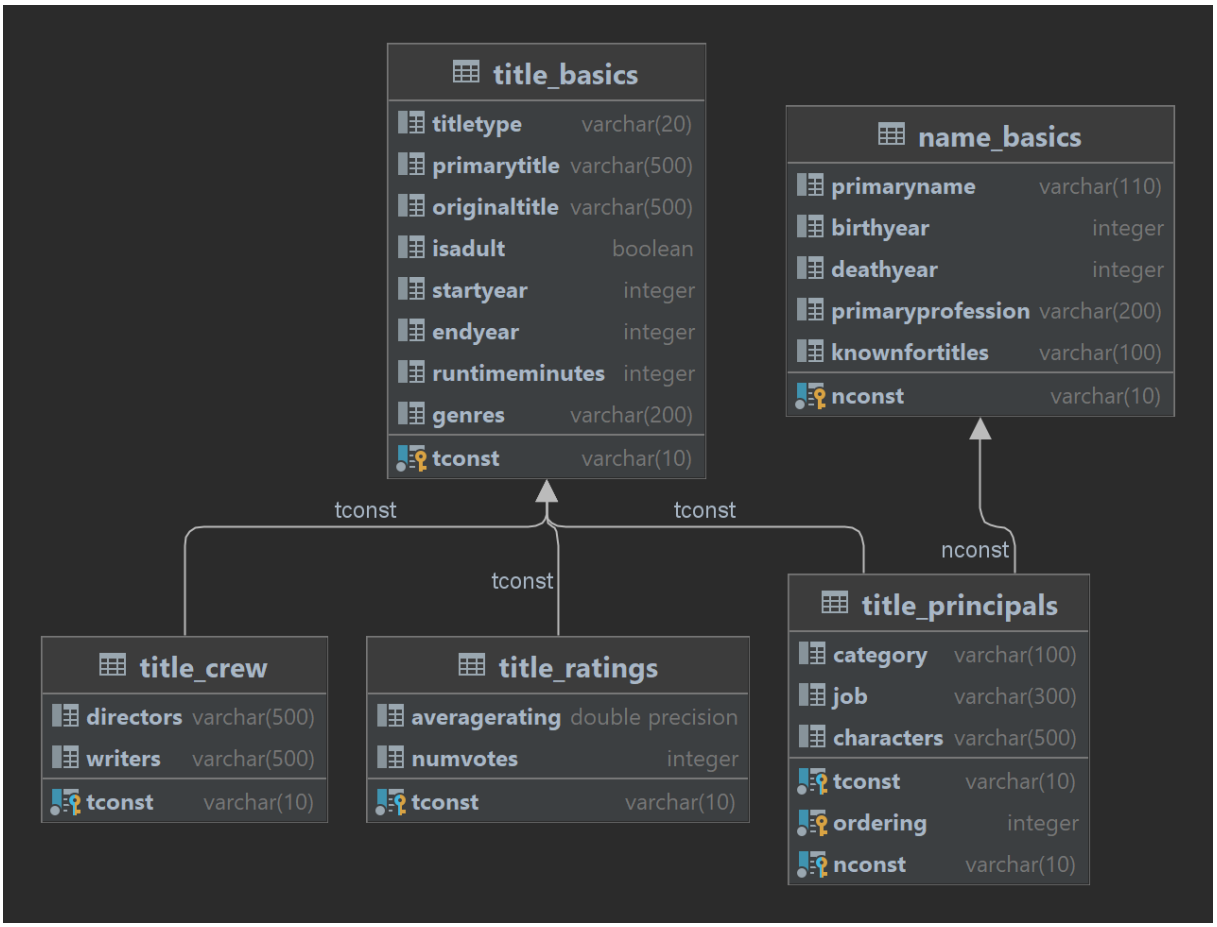
- Permettre à l'utilisateur d'effectuer une recherche d'un film par son titre principal ou son titre original
- Permettre à l'utilisateur de lister les films les plus populaires d'un genre

Décomposition

La base de données

Voici un schéma du contenu de la base de données qui est utilisée pour ce projet.

La base de données est PostgreSQL utilisée via docker.



Name Basics stocke les informations d'une personne

TitleBasics contient des informations basiques sur un titre

TitleRatings contient les avis et les notes d'un titre

TitlePrincipals contient les informations du casting et des équipes d'un titre

TitleBasics contient les informations concernant les directeurs et les scénaristes

Le fonctionnel

Le fonctionnel a pour but essentiel de mettre en place l'API qui permettra de répondre aux besoins du projet.

Les technologies

Quarkus : version 2.13.2. Final

La partie fonctionnelle du projet a été réalisée à l'aide de Quarkus.

Quarkus est un framework java conçu pour la compilation native, ceci le rend plus rapide d'exécution. Il est compatible avec des librairies java et des frameworks java populaires et se mesure à Spring, un framework bien établi.

Chaque entité étend de Panache. Panache est une librairie spécifique à Quarkus et permet d'éviter la répétition du code et permet de mettre en place le modèle Record Pattern.

Ce modèle permet de manipuler les objets de la classe et appliquer les méthodes usuelles de création, lecture, suppression et mise à jour de l'objet.

L'API

L'API est construite à l'aide de l'extension quarkus-resteasy-reactive-jackson.

Nous avons les routes suivantes :

/api/title-basics : renvoie 10 films

/api/title-basics/{titre} : renvoie les films dont le titre ou le titre original correspond entièrement ou partiellement à la recherche utilisateur

/api/title-basics/topratedmovies/{genre} : renvoie les films les mieux évalués selon un genre, correspondant entièrement ou partiellement à la recherche utilisateur.

L'interface

Les technologies

L'interface a été réalisée à l'aide d'Angular 14.2.6.

Le framework est bien documenté et est facile à prendre en main.

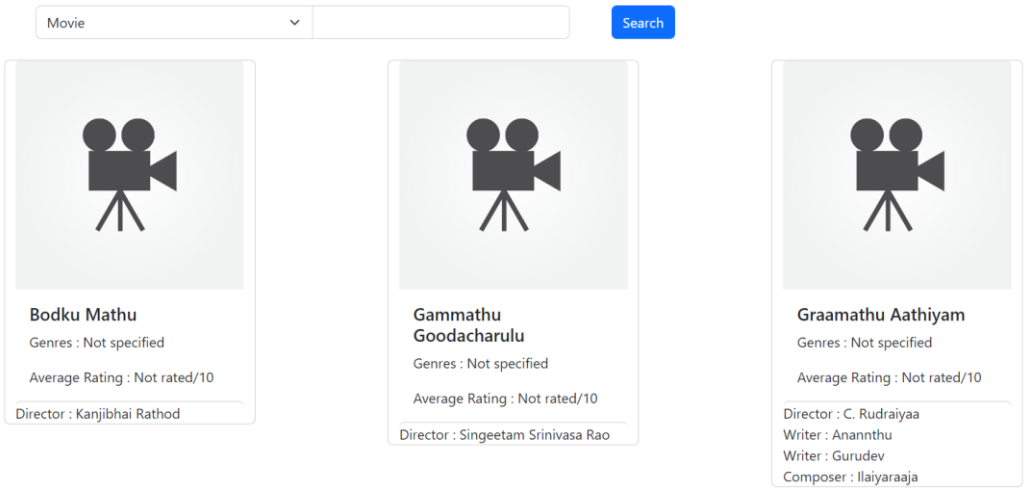
Le rôle

Elle permet de rechercher les films selon leur titre (movie) ou selon leur genre (genre). La casse n'est pas prise en compte.

Recherche par titre :

Il faudra sélectionner dans le menu déroulant « movie » et ensuite taper le nom du titre recherché,

Un maximum de 10 résultats sera affiché, c'est un choix arbitraire.



Recherche par genre :

Il faudra sélectionner dans le menu déroulant « genre » et ensuite taper le nom du genre recherché.

Les résultats seront affichés selon les notes données par les internautes.

Axes d'amélioration

Kevin Bacon Game

Route: /api/title-basics/ /baconNumber/{name}-{lastname}

L'idée du Kevin Bacon Game est de pouvoir lier un acteur à un autre via ses rôles dans les films.

Kevin Bacon a donc pour BaconNumber 0, si un acteur a joué avec lui dans un film il aura donc pour BaconNumber 1.

En me basant sur ce fait, j'ai recherché un algorithme efficace pour trouver le plus petit chemin entre l'acteur selectionné et Kevin Bacon.

L’objectif est de boucler dans la liste d’acteur, en partant de Kevin Bacon. Cette liste d’acteur est la liste d’acteurs qui ont joués dans un film avec Kevin Bacon. On assigne à chacun de ces acteurs un BaconNumber jusqu’à trouver la personne recherchée.

J’ai codé un service KevinBaconGame, celui-ci implémente une fonction getBaconNumber or je n’arrive pas à renvoyer le baconNumber, celui-ci me renvoie toujours la valeur par défaut.

NameBasicsConverter

J’ai utilisé un AttributeConverter nommé NameBasicsConverter afin de convertir les champs directors et writers des tableaux d’id de NameBasics en Liste de NameBasics.

Or j’ai une erreur quand j’utilise une ressource extérieure, ici en l’occurrence la fonction NameBasics.findByNConst dans le converter.

Le converter peut être utilisable plusieurs fois dans une entité car il étend de Repeatable, en l’occurrence, la ressource extérieure ne peut être appelée deux fois.