

Nom du projet : **MovieGraphInsights**

Objectif : Application Spring Boot qui exploite les données de la base Neo4j open source.

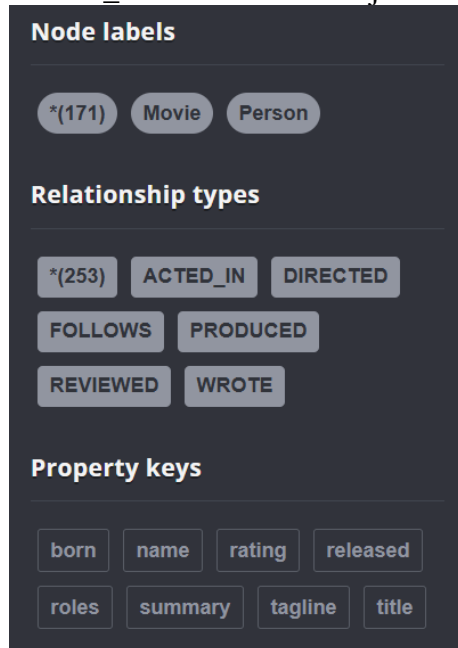
Base de données :

NEO4J_URI=bolt://3.83.163.178

NEO4J_USERNAME=neo4j

NEO4J_PASSWORD=handler-sill-generator

NEO4J_DATABASE=neo4j



Fonctionnalités principales

1. Analyse des relations entre films et acteurs

- Visualiser les co-acteurs fréquents sous forme de liste ou tableau simple.
- Filtrer par genre de film ou par année.

2. Statistiques clés

- Afficher le genre de films le plus populaire.
- Lister les acteurs les plus connectés (ceux qui ont collaboré avec le plus grand nombre d'autres acteurs).

3. API REST

- Fournir une API REST minimale avec les endpoints suivants :
 - /actors/connections: Liste des relations entre acteurs.
 - /movies/popular-genres: Genre de films le plus populaire.
 - /actors/most-connected: Acteurs les plus connectés.

Proposition d'Interfaces (JSP)

1. Page d'accueil

- Résumé des fonctionnalités disponibles.
- Liens vers les différentes sections (Relations, Statistiques, Documentation API).

2. Section "Relations Acteurs-Films"

- **Tableau des co-acteurs fréquents :**
 - Colonne 1 : Acteur principal.
 - Colonne 2 : Liste des co-acteurs.
- **Filtres dynamiques :**
 - Genre de film (menu déroulant).
 - Année de sortie (zone de texte).

3. Section "Statistiques"

- **Genre de films le plus populaire :**
 - Affichage simple (texte ou graphique statique).
- **Acteurs les plus connectés :**
 - Liste numérotée avec le nombre de collaborations.

4. Documentation API

- Description des endpoints REST disponibles avec des exemples d'URL.
- Formulaire de test simple pour effectuer des requêtes (champ pour saisir l'URL et afficher la réponse JSON).

Technologies et Stack

- **Backend :** Spring Boot (API + logique métier).
- **Base de données :** Neo4j.
- **Front-End :** JSP pour les pages dynamiques.
- **Styles :** Bootstrap ou CSS basique pour une mise en page propre.
- **Déploiement local :** Tomcat (avec Maven).