

# Répartition claire des rôles

## Partie MongoDB (Films)

Tout ce qui concerne les films.

### à faire

- Connexion à MongoDB
- Accès aux données films
- Endpoints liés aux films

### Fonctionnalités gérées

1. Lister tous les films
2. Rechercher un film (par titre ou acteur)
3. Mettre à jour un film
4. Récupérer la liste des films MongoDB (pour le croisement plus tard)

Fournir des fonctions réutilisables, pas seulement des routes.

## Partie Neo4j (Utilisateurs & notes)

Tout ce qui touche aux utilisateurs et aux relations.

### à faire

- Connexion à Neo4j
- Modèle User / Movie dans Neo4j
- Requêtes Cypher

### Fonctionnalités gérées

1. Lister les utilisateurs qui ont noté un film

2. Infos complètes d'un utilisateur (nb de films notés + liste)
3. Récupérer la liste des films Neo4j (pour le croisement)

## Structure de projet

```
backend/
|
├─ app/
|   └─ main.py                # Point d'entrée (FastAPI ou Flask)
|
|   └─ config/
|       └─ mongo.py          # Connexion MongoDB
|       └─ neo4j.py          # Connexion Neo4j
|
|   └─ routes/
|       └─ movies.py         # Routes MongoDB
|       └─ users.py          # Routes Neo4j
|       └─ stats.py          # Routes mixtes (à faire ensemble)
|
|   └─ services/
|       └─ movie_service.py  # Logique MongoDB
|       └─ user_service.py   # Logique Neo4j
|       └─ common_service.py# Croisement MongoDB / Neo4j
|
|   └─ models/
|       └─ movie.py
|       └─ user.py
|
├─ requirements.txt
└─ README.md
```