



UNIVERSITÉ SORBONNE SCIENCES JUSSIEU

PC3R : PROJET APPLI WEB

Etudiants :

Farouck Cherfi
Lina Azerouk

Table des matières

1 Projet	2
1.1 API Externe	2
1.2 Sujet	2
2 Dossier	3
2.1 Fonctionnalités	3
2.2 Cas d'utilisations	3
2.2.1 Cas d'utilisation Connexion	3
2.2.2 Cas d'utilisation Inscription	3
2.2.3 Cas d'utilisation Ajout Evaluation	3
2.3 Base de données	4
3 Serveur	4
3.1 ConnexionServlet	4
3.2 InscriptionServlet	4
3.3 MovieDBServlet	4
3.4 SeriesDBServlet	4
3.5 RechercheServlet	5
3.6 CommentairesServlet	5
3.7 CommentaireUtilisateurServlet	5
4 Client	5
4.1 Technologies clients	5
4.2 Ecrans	5
4.2.1 Ecran Home	5
4.2.2 Ecran Resultat Recherche	6
4.2.3 Ecran Connexion	6
4.2.4 Ecran Inscription	7
4.2.5 Ecran Profil	7
4.2.6 Ecran InformationsFilm	8
4.3 Requêtes et Réponses	8
4.3.1 Connexion	8
4.3.2 Inscription	8
4.3.3 Commentaire/Evaluation	8
4.3.4 Recherche	9
4.3.5 Films/Séries	9
4.3.6 Les commentaires/évaluation de l'utilisateur	9

1 Projet

1.1 API Externe

Nous avons choisi l'API externe MovieDB, qui est un service web RESTful, pour obtenir des informations sur des films et des séries.

Nous l'utiliserons principalement pour récupérer des informations sur des films et des séries.

Pour chaque film, l'API nous fournit les informations suivantes :

- Le titre : 'title'
- Le résumé : 'overview'
- L'image du poster : 'poster_path'
- La date de sortie : 'release_date'

Pour chaque série, l'Api nous fournit les informations suivantes :

- Le titre : 'name'
- Le résumé : 'overview'
- L'image du poster : 'poster_path'
- La date de sortie du premiere episode : 'first_air_date'

Cette API requiert une clé d'authentification qui peut être obtenue en s'inscrivant sur le site web de MovieDB. Depuis le 16 décembre 2019, il n'y a pas de limite d'appels sur l'API.

Par exemple, pour effectuer un appel à cette API afin de récupérer les films les plus populaires du moment :

```
final String API_KEY = "ded6000884579524d7f23a2993383461";  
"https://api.themoviedb.org/3/movie/popular?api_key="  
    + API_KEY  
    + "&language=fr-FR";
```

On recevra en réponse un objet au format JSON qui contient :

```
result : array[object]  
/* Pour chaque object */  
object : poster_path : string  
        overview : string  
        release_date : string  
        id : int  
        ...
```

1.2 Sujet

Nous avons choisi de créer un site de critiques et d'évaluations de films et séries.

2 Dossier

2.1 Fonctionnalités

Les principales fonctionnalité de notre application web sont les suivantes :

- Connexion
- Inscription
- Recherche par titre de série ou film
- Ajout d'un commentaire
- Ajout d'une évaluation
- Consultation de profil
- Suppression d'un commentaire
- Suppression d'une évaluation
- Déconnexion

2.2 Cas d'utilisations

2.2.1 Cas d'utilisation Connexion

Un utilisateur décide d'appuyer sur le bouton "Connexion". Il est alors redirigé vers la page de connexion où il peut saisir son identifiant et son mot de passe. Si l'utilisateur saisit un identifiant et un mot de passe correspondant à la base de données, il est redirigé vers la page d'accueil. Dans le cas contraire, un message d'erreur s'affichera pour indiquer que l'identifiant ou le mot de passe est incorrect.

2.2.2 Cas d'utilisation Inscription

Lorsqu'un utilisateur décide d'appuyer sur le bouton "Inscription", il est redirigé vers la page d'inscription où il peut saisir son identifiant et son mot de passe. Si l'utilisateur saisit un identifiant déjà existant dans la base de données, un message d'erreur s'affichera pour lui indiquer que l'identifiant est déjà utilisé. Dans le cas contraire, il sera redirigé vers la page de connexion après avoir terminé le processus d'inscription.

2.2.3 Cas d'utilisation Ajout Evaluation

Lorsqu'un utilisateur décide d'ajouter une évaluation à un film, il peut cliquer sur le film "Mario Bros" et ajouter son commentaire. Si l'utilisateur est connecté, son évaluation sera affichée sur la page du film. Dans le cas contraire, il sera redirigé vers la page de connexion. Si l'utilisateur avait déjà ajouté une évaluation pour ce film, sa note sera mise à jour.

2.3 Base de données

Nous utilisons une base de données de type NoSQL pour stocker nos données. La base de données est mise à jour à chaque appel à l'API externe, qui est elle-même mise à jour par l'utilisateur. Les données peuvent être modifiées par l'utilisateur lorsqu'il crée un compte, ajoute un commentaire ou une évaluation.

Voici quelques schémas de tables :

Movies :

- id : *int*
- titre : *string*
- resume : *string*
- poster : *string*
- date : *date*
- evaluations : *int*
- commentaires : *int*

Utilisateurs :

- identifiant : *string*
- mdp : *tokken*
- commentaires : *int*
- evaluations : *int*

Commentaires et évaluations correspondent respectivement aux nombres de commentaires et d'évaluations.

3 Serveur

Nous avons mis en place une API REST pour notre serveur en adoptant une approche basée sur les ressources et d'une approche classique.

3.1 ConnexionServlet

Cette servlet permet de vérifier si l'identifiant et le mot de passe appartiennent à la base de données. Elle dispose d'une méthode POST qui renvoie un fichier JSON.

3.2 InscriptionServlet

Cette servlet permet d'ajouter un utilisateur dans la base de données. Elle dispose d'une méthode POST qui renvoie un fichier JSON.

3.3 MovieDBServlet

Cette servlet permet de faire un appel à l'API externe movieDB et récupère les films populaires du moment. Elle possède une méthode GET qui renvoie un fichier JSON.

3.4 SeriesDBServlet

Cette servlet permet de faire un appel à l'API externe MovieDB et de récupérer les séries populaires du moment. Elle possède une méthode GET qui renvoie un fichier JSON.

3.5 RechercheServlet

Cette servlet permet de récupérer les films et les séries de la base de données avec un certain nom de recherche. Elle possède une méthode POST qui renvoie un fichier JSON.

3.6 CommentairesServlet

Cette servlet permet de gérer les commentaires relatifs à un film de la base de données. Elle possède des méthodes POST, GET et DELETE.

3.7 CommentaireUtilisateurServlet

Cette servlet permet de récupérer les commentaires relatifs à un utilisateur. Elle possède la méthode GET.

Les servlets EvaluationServlet et EvaluationUtilisateurServlet font les même chose que les servlets CommentairesServlet et CommentaireUtilisateurServlet.

4 Client

4.1 Technologies clients

Nous avons utilisé Bootstrap et React comme technologies pour notre application web. Nous avons choisi Bootstrap pour sa facilité d'utilisation pour créer un header et une barre de recherche. Nous avons opté pour les composants React pour leur flexibilité et leur capacité à rendre notre application plus interactive.

4.2 Ecrans

4.2.1 Ecran Home

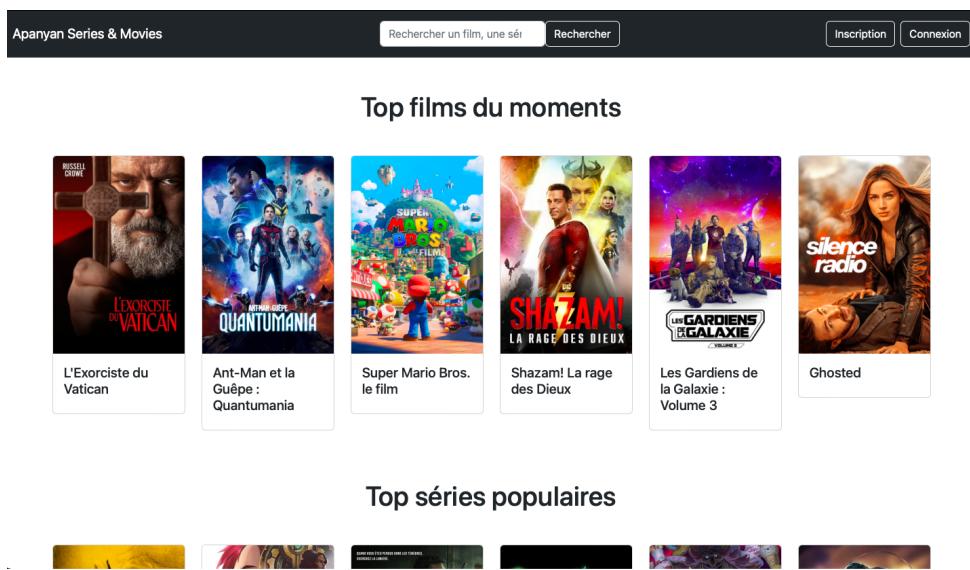


FIGURE 1 – Ecran Home

4.2.2 Ecran Resultat Recherche

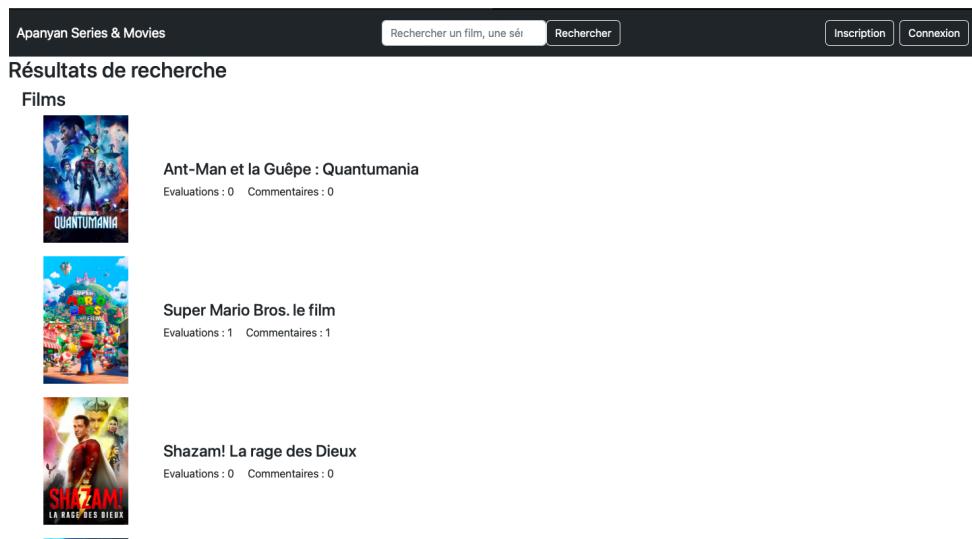


FIGURE 2 – Ecran Resultat

4.2.3 Ecran Connexion

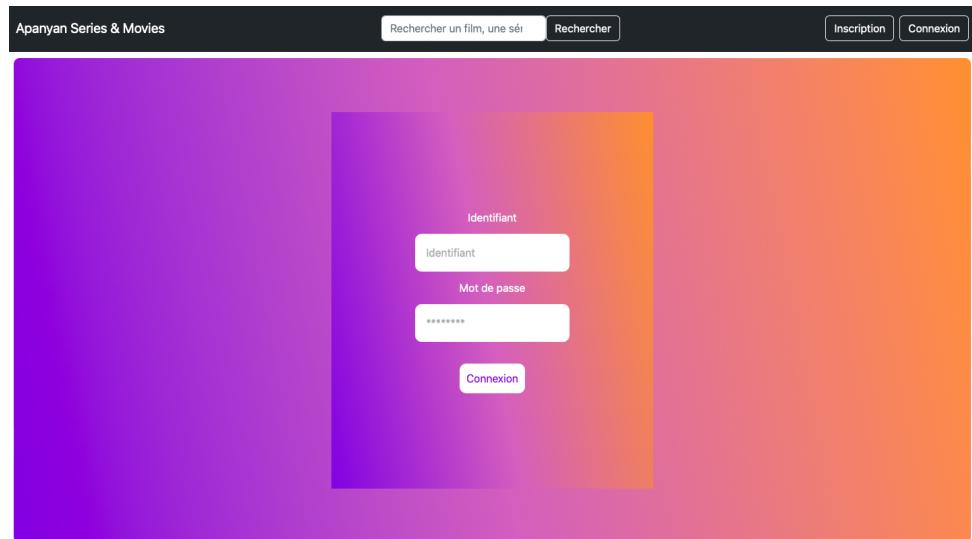


FIGURE 3 – Ecran Connexion

4.2.4 Ecran Inscription

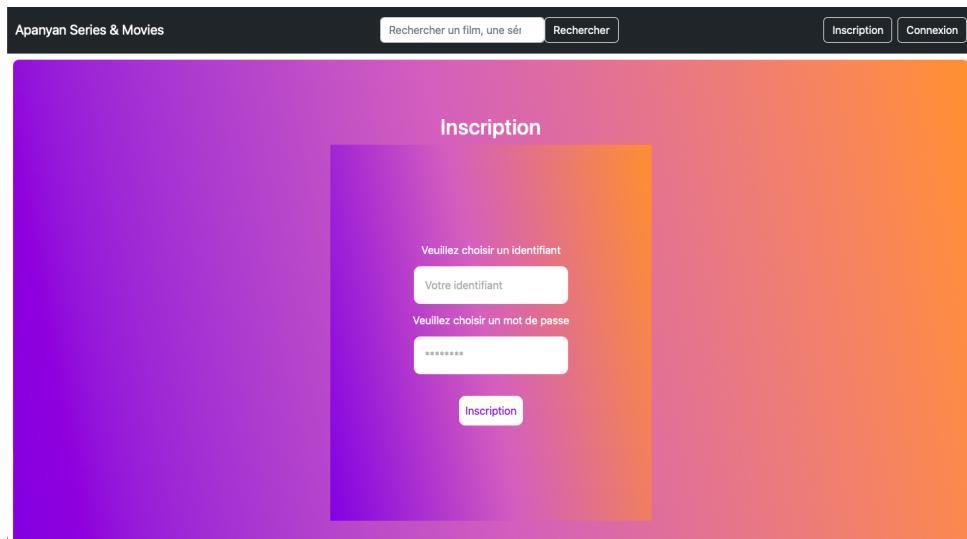


FIGURE 4 – Ecran Inscription

4.2.5 Ecran Profil

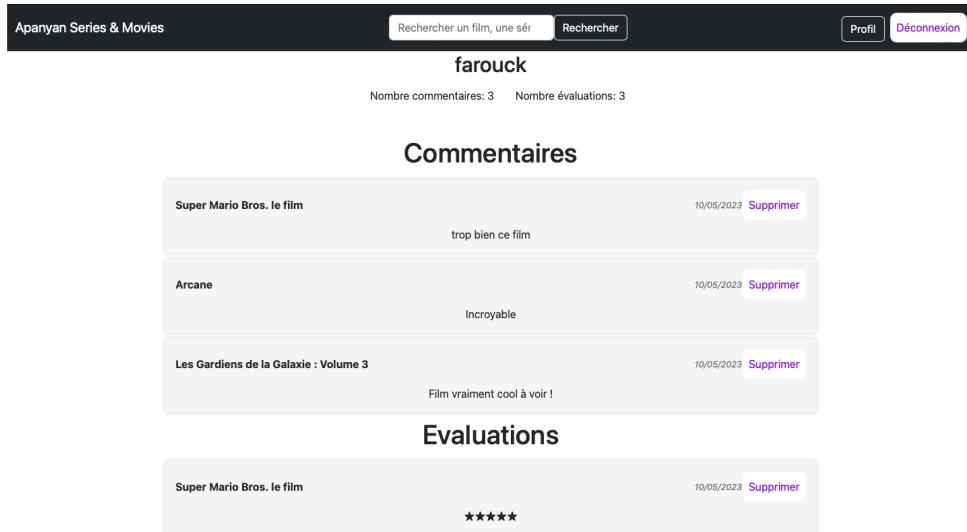


FIGURE 5 – Ecran Profil

4.2.6 Ecran InformationsFilm

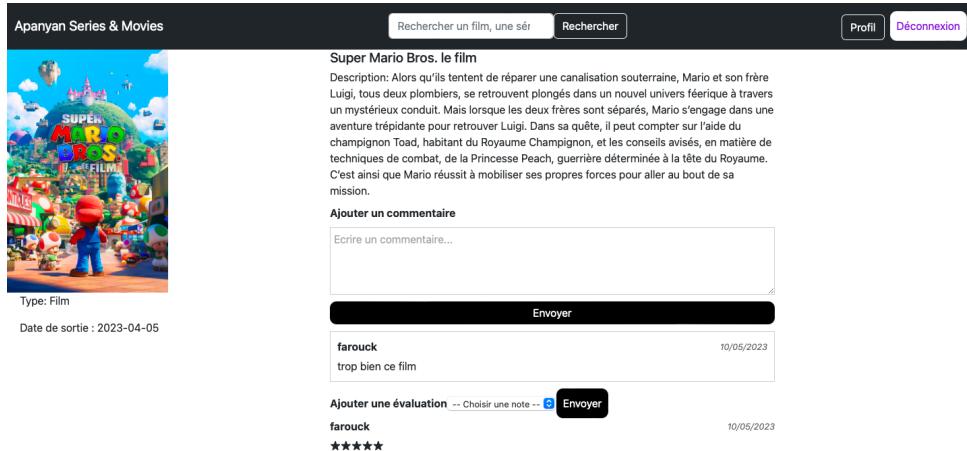


FIGURE 6 – Ecran InformationsFilm

4.3 Requêtes et Réponses

Toutes les requêtes sont effectuées en utilisant le protocole HTTP, et les réponses sont envoyées sous forme de JSON.

4.3.1 Connexion

doPost identifiant_utilisateur password \implies statut : bool, message : string, utilisateur : informationsuser Statut : True si la connexion a réussi et renvoie les données de l'utilisateur au client, sinon il envoie un message d'erreur.

4.3.2 Inscription

doPost identifiant_utilisateur password \implies statut : bool, message : string, utilisateur : informationsuser Statut : True si l'inscription a réussi et renvoie les données de l'utilisateur créé au client, sinon il envoie un message d'erreur.

4.3.3 Commentaire/Evaluation

Recupère les commentaires/évaluations doGet idfilm \implies résultats : listcommentaires ou listevaluations

Ajoute un commentaire/évaluation doPost idfilm type_id identifiant_utilisateur commentaire \implies statut : bool, message : string Message : Ajout de commentaire ou d'évaluation OK, sinon message d'erreur.

Supprime un commentaire/évaluation doDelete type idfilm identifiant_utilisateur \implies statut : bool, message : string Message : Suppression de commentaire ou évaluation OK, sinon message d'erreur.

4.3.4 Recherche

doGet titre_recherche \Rightarrow resultats : listfilms et list series qui ont un mot/lettre correspondant au regex de la recherche.

4.3.5 Films/Séries

doGet \Rightarrow resultats : listfilms populaires du moment ou listseries populaires

4.3.6 Les commentaires/évaluation de l'utilisateur

doGet identifiant_utilisateur \Rightarrow resultats : listcommentaires ou listevaluations de l'utilisateur