Table of contents

La réservation de scéances de cinéma

Contexte

Notre site de consultation de films ayant du succès, nous souhaiterions désormais permettre aux utilisateurs de réserver leurs scéance via notre plateforme. Nous souhaitons ainsi la réalisation d'un nouveau webservice de réservation et renforcer notre infrastructure actuelle.

Contenu de ce document

Ce document contient les contraintes que nous devons respecter en terme d'infrastructure et de service `movie` et `reservation`. `authentification` ne fait pas partie du périmètre et ne devra pas être modifié. Au delà des contraintes exprimés dans ce documents, vous pourrez effectuer les modifications dont vous avez besoin.

Infrastructure

Il sera attendu la possibilité de gérer une forte charge lors des sorties des nouvelles scéances. Nous souhaiterions ainsi une infrastructure avec la séparation suivante :-

- Base de données
- Service d'identification
- Service de films
- Service de réservation

Il se pose aussi la question de la pertinence de la gestion d'une job queue avec RabbitMQ ou service équivalent.

Le webservice de films

Les routes

Il devra respecter le contrat d'interface suivant :

Verbe	Chemin	Entrée	Sortie	Description
GET	/movies		[{uid, name, description, rate, duration, hasReservationsAvailable, createdAt, updatedAt}]	liste les films enregistrer, il permettra d'affichier si des réservations sont disponible ou non
GET	/movies/{uid}		{uid, name, description, rate, duration, hasReservationsAvailable, createdAt, updatedAt}	Permet de récupérer un film
POST	/movies	{uid, name, description, rate, duration}	{uid, name, description, rate, duration, hasReservationsAvailable, createdAt, updatedAt}	Permet de créer un film
PUT	/movies/{uid}	{uid, name, description, rate, duration}	{uid, name, description, rate, duration, hasReservationsAvailable, createdAt, updatedAt}	Permet de modifier un film
DELETE	/movies/{uid}			Permet de supprimer un film

L'objet movie

L'objet movie devra respecter les contraintes suivantes :

- uid : un **UUID V4** pour le format
- name : une chaine de caractère de maximum **128 charactères ne pouvant être vide**
- description : un texte de maximum **4096 charactères ne pouvant être vide**
- rate : un entier ayant une valeur entre **1 et 5, ne pouvant être vide**
- duration : un entier, supérieur à 0, **inférieur à 240, ne pouvant être vide**

Les retours

Les retours seront les suivants :

- GET /movies :
  - 200 : les films sont listé avec succès
  - 204 : Pas de résultat de recherche
  - 400 : Les paramètres de recherche sont invalide
  - 404 : La page n'existe pas
  - 500 : erreur interne
- GET /movies/{uid}} :
  - 200 : Le film est affiché avec succès
  - 404 : Le film est inconnu
  - 500 : erreur interne
- POST /movies :
  - 201 : Le film est créé avec succès
  - 422 : Le contenu de l'objet film dans le body est invalide
  - 500 : erreur internew
- PUT /movies/{uid}} :
  - 200 : Le film est mis à jour avec succès
  - 404 : Le film est inconnu

    - 422 : Le contenu de l'objet film dans le body est invalide
    - 500 : erreur interne
- DELETE /movies/{uid}} :
    - 204 : Les film est affiché avec succès
    - 404 : Le film est inconnu
    - 500 : erreur interne

Les paramètres additionnels

Il sera possible d'effectuer des recherches de films à l'aide du paramètre query : ex: `GET /movies?query=wall-e`

La recherche sera effectuée dans le titre ainsi que dans la description.

Le webservice de réservation

Description

Ce webservice permettra d'effectuer des réservations dans une salle de cinéma. Les utilisateurs pourront réservés le nombre de places qu'ils souhaitent.

Les routes

Il devra respecter le contrat d'interface suivant :

Verbe	Authorisations	Chemin	Entrée	Sortie	Description
POST	Client	/movie/{movieUid}/reservations	{sceance, nbSeats, room}	{uid, rank, status, createdAt, updatedAt, expiresAt}	Permet de rentrer dans le tunnel de réseration
POST	Client	/reservations/{uid}/confirm			Permet de confirmer la réservation si le status le permet
GET	Admin	/movie/{movieUid}/reservations		[{uid, rank, status, createdAt, updatedAt, expiresAt}]	Liste toutes les réservations en cours pour un film
GET	Admin|Owner	/reservations/{uid}		{uid, rank, status, createdAt, updatedAt, expiresAt}	Permet de récupérer le détail d'une réservation
GET	Client	/cinema		[{uid, name, createdAt, updatedAt}]	Permet de lister les cinéma
GET	Client	/cinema/{uid}		{uid, name, createdAt, updatedAt}	Permet de créer un cinéma
POST	Admin	/cinema	{uid, name}	{uid, name, createdAt, updatedAt}	Permet de créer un cinéma
PUT	Admin	/cinema/{uid}	{uid, name}	{uid, name, createdAt, updatedAt}	Permet de modifier un cinéma
DELE	Admin	/cinema/{uid}			Permet de supprimer un cinéma

TE					
GET	Client	/cinema/{cinemaUid}/rooms		[{uid, seats, name, createdAt, updatedAt}]	Permet de lister les salles de cinéma
GET	Client	/cinema/{cinemaUid}/rooms/{uid}		{uid, seats, name, createdAt, updatedAt}	Permet d'afficher une salle de cinéma
POST	Admin	/cinema/{cinemaUid}/rooms		{uid, seats, name, createdAt, updatedAt}	Permet de créer une salle de cinéma
PUT	Admin	/cinema/{cinemaUid}/rooms/{uid}		{uid, seats, name, createdAt, updatedAt}	Permet de modifier une salle de cinéma
DELETE	Admin	/cinema/{cinemaUid}/rooms/{uid}			Permet de supprimer une salle de cinéma
GET	Client	/cinema/{cinemaUid}/rooms/{roomUid}/sceances		[{uid, movie, date}]	Permet de lister les scéances d'une salle
POST	Admin	/cinema/{cinemaUid}/rooms/{roomUid}/sceances	{movie, date}	{uid, movie, date}	Permet de créer une scéance dans une salle
PUT	Admin	/cinema/{cinemaUid}/rooms/{roomUid}/sceances/{uid}	{movie, date}	{uid, movie, date}	Permet de modifier scéance dans une salle
DELE	Admin	/cinema/{cinemaUid}/rooms/{roomUi			Permet de supprimer une scéance da

TE		d}/sceances/{uid}			ns une salle

L'objet cinéma

L'objet cinéma devra respecter les contraintes suivantes :

- uid : un **UUID V4** pour le format
- name : une chaine de caractère de maximum **128 charactères ne pouvant être vide**

L'objet room devra respecter les contraintes suivantes :

- uid : un **UUID V4** pour le format
- name : une chaine de caractère de maximum **128 charactères ne pouvant être vide**
- seats : **un entier, supérieur à 0, ne pouvant être vide**

L'objet reservation devra respecter les contraintes suivantes :

- uid : un **UUID V4** pour le format
- rank : un **entier** pour le format, ne peut être inferieur à 0, ne peut être nulle
- status : devra être une des valeurs suivantes :
  - open: il est possible de confirmer sa réservation
  - expired: la réservation n'est plus possible
  - confirmed: la réservation est terminé
- seats : **un entier, supérieur à 0, ne pouvant être vide**

L'objet sceance devra respecter les contraintes suivantes :

- uid : un **UUID V4** pour le format
- movie : un **UUID V4** pour le format, correspond à l'UID d'un film
- date : une date de scéance

Les retours

Les retours seront les suivants :

- POST /movie/{movieUid}/reservations :
  - 201 : L'entrée dans de tunnel de réservation est un succès
  - 422 : Le contenu de l'objet reservation dans le body est invalide
  - 500 : erreur internew
- POST /reservations/{uid}/confirm
  - 201 : Réservation effectuée avec succès
  - 422 : Le contenu de l'objet reservation dans le body est invalide
  - 410 : La réservation est expirée
  - 500 : erreur internew
- GET /movie/{movieUid}/reservations
  - 200 : Les réservations sont affichés avec succès
  - 404 : Le film est inconnu
  - 500 : erreur interne
- GET /reservations/{uid}
  - 200 : La réservation est affiché avec succès
  - 404 : La réservation est inconnu
  - 500 : erreur interne
- GET /cinema :
  - 200 : les cinémas sont listé avec succès
  - 204 : Pas de résultat de recherche
  - 400 : Les paramètres de recherche sont invalide

    - 404 : La page n'existe pas
    - 500 : erreur interne
- GET /cinema/{uid}} :
    - 200 : Le cinéma est affiché avec succès
    - 404 : Le cinéma est inconnu
    - 500 : erreur interne
- POST /cinema :
    - 201 : Le cinéma est créé avec succès
    - 422 : Le contenu de l'objet cinema dans le body est invalide
    - 500 : erreur internew
- PUT /cinema/{uid}} :
    - 200 : Le cinéma est mis à jour avec succès
    - 404 : Le cinéma est inconnu
    - 422 : Le contenu de l'objet cinema dans le body est invalide
    - 500 : erreur interne
- DELETE /cinema/{uid}} :
    - 204 : Les cinema est affiché avec succès
    - 404 : Le cinema est inconnu
    - 500 : erreur interne

Le fonctionnement général

L'utilisateur devra prendre un ticket en s'inscrivant dans une liste d'attente avec la route `POST /movie/{movieUid}/reservations` il aura les informations suivantes :

- sa place dans la file
- un status indiquant si il peut valider son inscription ou si sa demande est expiré

- une date d'expiration de la demande si c'est à son tour

Une fois que c'est au tour de l'utilisateur de valider son inscription, il pourra la valider avec `POST /reservations/{uid}/confirm` si sa place n'est pas expiré.

Les mails

L'utilisateurs recevra deux types de mails :

- une communication lors de la création de nouvelles scéances
- Un mail de confirmation une fois la scéance réservé