

Description app.js

Ce code met en place un serveur backend avec Express, un framework Node.js largement utilisé pour créer des API. Il commence par importer les modules nécessaires : `express` pour le serveur, `cors` pour gérer les requêtes provenant d'autres origines (même si le commentaire indique que ce module n'est pas indispensable pour l'instant), et `dotenv` pour charger les variables d'environnement depuis un fichier `.env`.

L'application Express est instanciée avec `const app = express()`. Ensuite, `app.use(cors())` active la gestion CORS, ce qui permet à des applications frontend hébergées sur d'autres domaines d'accéder à l'API. La ligne `app.use(express.json())` permet à l'application de comprendre et de traiter automatiquement les requêtes HTTP contenant des données au format JSON, ce qui est essentiel pour la plupart des API REST modernes.

Le code importe ensuite quatre modules de routes, chacun correspondant à une fonctionnalité spécifique de l'application : `concerts`, `utilisateurs`, `réservations` et `accompagnements`. Ces modules sont ensuite associés à des chemins d'URL spécifiques grâce à `app.use()`. Par exemple, toutes les requêtes commençant par `/api/concerts` seront redirigées vers le module `concertRoutes`.

Enfin, le serveur écoute sur un port défini par la variable d'environnement `PORT`, ou sur le port 3000 par défaut. Lorsqu'il démarre, il affiche un message dans la console pour indiquer que le serveur est prêt à recevoir des requêtes. Ce fichier constitue donc le point d'entrée principal de l'application backend, centralisant la configuration et la gestion des différentes routes de l'API.

Pseudo code

Importer le module express

Importer le module cors

Charger les variables d'environnement avec dotenv

Créer une application express

Activer le middleware cors

Activer le middleware pour traiter les requêtes JSON

Importer les modules de routes pour concerts, utilisateurs, réservations et accompagnements

Associer chaque module de routes à son chemin d'API respectif

Définir le port d'écoute (depuis les variables d'environnement ou 3000 par défaut)

Démarrer le serveur et afficher un message dans la console quand il est prêt.

Méthodes

- **require()** : pour importer des modules Node.js ou des fichiers locaux.
- **config()** : méthode du module dotenv pour charger les variables d'environnement.
- **express()** : pour créer une instance de l'application Express.
- **use()** : méthode d'Express pour ajouter des middlewares ou des routes à l'application.
- **listen()** : méthode d'Express pour démarrer le serveur et écouter sur un port donné.
- **log()** : méthode de console pour afficher un message dans la console.