

Fichier : 01 - app.js

Tutoriel Node.js avec Express, CORS et Dotenv

Ce tutoriel explique comment mettre en place un serveur Node.js simple en utilisant les bibliothèques Express, CORS et Dotenv. Nous allons également définir des routes et des services spécifiques pour une API d'intelligence artificielle (AI).

Importation des Modules

```
import express from 'express';
import cors from 'cors';
import dotenv from 'dotenv';

import aiRoutes from './routes/ai.js';
import aiServices from './config/ai-services.js';
```

Dans ce bloc de code, nous importons les modules nécessaires pour notre application :

- `express` : un framework pour Node.js qui simplifie le développement d'applications web.
- `cors` : un package Node.js pour fournir un middleware Connect/Express qui peut être utilisé pour activer CORS (Cross-Origin Resource Sharing) avec diverses options.
- `dotenv` : un module qui charge les variables d'environnement à partir d'un fichier `.env` dans `process.env`.
- `aiRoutes` : un module définissant les routes pour notre API d'AI.
- `aiServices` : un module définissant les services disponibles pour notre API d'AI.

Configuration de l'Application

```
dotenv.config();

const app = express();
const port = 3000;

app.use(cors());
app.use(express.json());
```

Ici, nous configurons notre application :

- Nous appelons `dotenv.config()` pour charger les variables d'environnement à partir d'un fichier `.env`.
- Nous initialisons une nouvelle application Express et définissons le port sur lequel notre serveur écoutera.
- Nous utilisons le middleware `cors()` pour activer CORS sur notre serveur.
- Nous utilisons `express.json()` pour analyser les corps des requêtes entrantes au format JSON.

Definition des Routes

```
app.use('/api/ai', aiRoutes);

app.get('/api/ai/services', (req, res) => {
  res.json({ services: aiServices });
});
```

Dans ce bloc, nous définissons les routes pour notre application :

- Nous utilisons `app.use()` pour monter le routeur `aiRoutes` sur le chemin `/api/ai`.
- Nous définissons une route GET a `/api/ai/services` qui renvoie la liste des services d'AI disponibles.

Demarrage du Serveur

```
app.listen(port, () => {
  console.log(`Server listening on http://localhost:${port}`);
});
```

Enfin, nous démarrons notre serveur en appelant `app.listen()`. Une fois que le serveur est prêt, il affiche un message indiquant qu'il écoute sur le port spécifié.