

Fichier : 01 - app.js

Tutoriel Node.js avec Express, CORS et dotenv

Ce tutoriel explique comment configurer un serveur Node.js en utilisant le framework Express, en ajoutant une gestion des CORS (Cross-Origin Resource Sharing) et en utilisant dotenv pour gerer les variables d'environnement.

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 cette section, nous importons les modules necessaires pour notre application.

- `express` est un framework pour Node.js qui simplifie le developpement de serveurs web.
- `cors` est un package Node.js qui fournit un middleware pour activer les CORS avec diverses options.
- `dotenv` est un module qui charge les variables d'environnement a partir d'un fichier `.env` dans `process.env`.
- `aiRoutes` est un module personnalise qui contient les routes pour notre API.
- `aiServices` est un module personnalise qui contient la configuration de nos services d'IA.

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 commencons par charger les variables d'environnement a l'aide de `dotenv.config()`.
- Nousinstancions ensuite notre application Express et definissons le port sur lequel notre serveur ecouterait.
- Nous ajoutons ensuite le middleware CORS a notre application pour permettre les requetes cross-origin.
- Enfin, nous utilisons `express.json()` pour analyser les corps des requetes entrantes dans un middleware avant vos questionnaires, disponibles sous la propriete `req.body`.

Definition des routes

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

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

Dans cette section, nous définissons les routes pour notre API.

- Nous utilisons `app.use` pour ajouter le middleware `aiRoutes` à notre application. Toutes les requêtes commençant par `/api/ai` utiliseront ce middleware.
- Nous définissons ensuite une route GET pour `/api/ai/services` qui renvoie un objet JSON contenant nos services d'IA.

Demarrage du serveur

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

Enfin, nous démarrons notre serveur en écoutant sur le port spécifié. Une fois que le serveur est démarré, un message est affiché dans la console pour indiquer que le serveur est en écoute.