

Pseudo code

Importer la classe Pool du module pg

Charger les variables d'environnement avec dotenv

Créer un pool de connexions PostgreSQL avec les paramètres récupérés depuis les variables d'environnement :

- hôte
- port
- nom de la base de données
- utilisateur
- mot de passe

Exporter ce pool pour qu'il soit utilisable dans d'autres fichier de l'application.

Description

Ce code configure la connexion à une base de données PostgreSQL en utilisant le module pg pour Node.js. Il commence par importer la classe Pool depuis le module pg, qui permet de gérer un ensemble de connexions réutilisables à la base de données (appelé « pool de connexions »). Cela améliore les performances de l'application en évitant d'ouvrir et de fermer une connexion à chaque requête.

La ligne `require('dotenv').config()` charge les variables d'environnement définies dans un fichier `.env`, ce qui permet de stocker de façon sécurisée les informations sensibles comme l'hôte, le port, le nom de la base, l'utilisateur et le mot de passe. Ces variables sont ensuite utilisées pour configurer le pool de connexions lors de sa création avec `new Pool({...})`.

Enfin, le pool ainsi configuré est exporté avec `module.exports = pool`. Cela permet à d'autres fichiers de l'application d'utiliser ce pool pour exécuter des requêtes SQL sur la base de données, en profitant de la gestion automatique des connexions offerte par la classe Pool.

Méthodes

- **require()** : pour importer des modules Node.js ou des fichiers locaux.
- **config()** : méthode du module dotenv pour charger les variables d'environnement.
- **Pool()** : constructeur de la classe Pool du module pg, utilisé pour créer un pool de connexions à PostgreSQL.
- **module.exports** : (ce n'est pas une méthode mais une syntaxe Node.js) pour exporter le pool et le rendre accessible dans d'autres fichiers.

Accolades

Les accolades autour de `Pool` dans `const { Pool } = require('pg');` signifient que l'on utilise **la décomposition (destructuring)** pour importer uniquement la classe `Pool` exportée par le module `pg`.

Le module `pg` exporte plusieurs éléments (par exemple : `Client`, `Pool`, etc.). En utilisant les accolades, tu récupères spécifiquement la classe `Pool` sans importer tout le contenu du module dans une seule variable. C'est une manière concise d'accéder directement à ce dont tu as besoin.