

Exercice 1 : Middleware de journalisation (Facile)

Objectif : Crée un middleware qui journalise les informations sur chaque requête entrante, telles que la méthode HTTP (GET, POST, etc.) et l'URL de la requête.

Instructions :

Crée un serveur Express simple.

Implémente un middleware qui journalise dans la console chaque fois qu'une requête est faite au serveur (affiche la méthode et l'URL de la requête).

Teste-le en effectuant différentes requêtes sur ton serveur (par exemple, GET, POST, etc.).

Bonus (difficile) :

L'idée est de créer un dossier de journalisation pour y avoir un fichier journalier qui recueille ce qui s'est passé jour par jour et notamment s'il y a eu des erreurs. Ci joint les grandes lignes et ensuite les étapes détaillées si tu as besoin d'être plus guidé.

Les grandes lignes

- Crée un serveur Express simple.
- - Installe et utilise la bibliothèque fs (File System) de Node.js pour manipuler les fichiers.
- Utilise le format de date YYYY-MM-DD pour le nom du fichier log.
- Assure-toi que le dossier log est créé si ce n'est pas déjà le cas (si le dossier existe, on peut directement y écrire).
- Crée un middleware pour capturer et écrire dans les fichiers log.
- Crée un middleware de gestion des erreurs pour capturer les erreurs et les enregistrer dans le log.

Étapes détaillées

Initialisez une application Express.

Importez les modules express, fs, et path.

Configurez un serveur Express simple.

Créez un dossier pour stocker les journaux.

Utilisez path pour définir un chemin vers un dossier appelé log.

Vérifiez si le dossier existe déjà avec fs.existsSync.

Si le dossier n'existe pas, créez-le avec fs.mkdirSync.

Ajoutez un middleware pour journaliser les requêtes.

Ajoutez un middleware global qui affiche dans la console la méthode HTTP et l'URL de chaque requête.

Créez une fonction nommée logToFile pour écrire les journaux dans des fichiers.

Enregistrez chaque requête dans un fichier log, avec la date, l'heure, la méthode, et l'URL.

Ajoutez des routes simples pour tester l'application.

Ajoutez une route GET /home qui renvoie "Accueil".

Ajoutez une route POST /api/user qui renvoie "Utilisateur créé".

Ajoutez une route GET /error qui lève une erreur volontaire pour tester la gestion des erreurs.

Ajoutez un middleware pour gérer les routes non trouvées (404).

Loguez les erreurs 404 dans un fichier log avec la méthode HTTP, l'URL, et un message "Route non trouvée".

Retournez une réponse JSON indiquant que la route n'existe pas.

Ajoutez un middleware pour gérer les erreurs (500).

Loguez les erreurs internes dans un fichier log avec leur message et les détails de la requête.

Retournez une réponse JSON avec un message d'erreur générique.

Démarrez le serveur.

Configurez l'application pour écouter sur le port 3000.

Affichez un message dans la console indiquant que le serveur est prêt.