

Énoncé du Travail Pratique

Titre : Création d'une API RESTful pour gérer des utilisateurs avec Node.js et Express

Objectif :

L'objectif de ce travail est de mettre en pratique vos compétences en gestion de modules dans Node.js, la création de serveurs, l'utilisation d'Express.js pour gérer des routes CRUD, ainsi que l'intégration d'API externes avec Axios.

Contexte

Vous allez développer une API simple pour gérer les utilisateurs d'une application. Chaque utilisateur aura les propriétés suivantes : id, name, et email. Vous utiliserez Node.js et Express pour créer cette API, et vous intégrerez également une API externe pour récupérer des informations supplémentaires sur les utilisateurs.

Tâches à Réaliser

1. Créer un projet Node.js :

- Initialisez un nouveau projet Node.js dans un répertoire de votre choix.
- Installez Express et Axios .

2. Créer une API RESTful :

- Définir un tableau d'utilisateurs :
 - Créez un tableau d'objets représentant des utilisateurs avec des propriétés id, name, et email.
- Implémenter les routes CRUD :
 - **GET /api/users** : Récupérer tous les utilisateurs.
 - **GET /api/users/:id** : Récupérer un utilisateur par son ID.
 - **POST /api/users** : Ajouter un nouvel utilisateur. (Le corps de la requête doit contenir name et email).
 - **PUT /api/users/:id** : Mettre à jour un utilisateur existant par son ID. (Le corps de la requête doit contenir les nouvelles valeurs pour name et email).
 - **DELETE /api/users/:id** : Supprimer un utilisateur par son ID.

3. Intégrer une API Externe avec Axios :

- Ajoutez une route **GET /api/users/:id/details** qui utilise Axios pour récupérer des informations supplémentaires sur un utilisateur à partir d'une API externe (par exemple, une API de gestion d'utilisateurs).
- Vous pouvez utiliser une API publique comme [JSONPlaceholder](#) pour simuler la récupération de données.

4. Tester l'API :

- Utilisez Postman pour tester chaque route de votre API.
- Vérifiez que les réponses sont correctes et que les erreurs sont gérées correctement (par exemple, utilisateur non trouvé).