

Exercice Ludique : Créer un Petit Serveur Web avec Node.js

Contexte

Bienvenue à **Magica**, une école fictive de magie où vous avez pour mission de créer un petit site web grâce au module `http` de Node.js. Les utilisateurs peuvent accéder à la page d'accueil et à une page "À propos". En plus de cela, nous allons vous demander de gérer des requêtes POST pour que les utilisateurs puissent soumettre leurs informations et recevoir une réponse personnalisée.

L'objectif de cet exercice est de vous familiariser avec la création de serveurs HTTP, la gestion des routes, et le traitement des requêtes GET et POST avec Node.js.

Objectifs de l'exercice

1. Créer un serveur web avec Node.js.
2. Gérer les routes HTTP (accueil et page "À propos").
3. Traiter une requête POST pour soumettre des données utilisateur.
4. Envoyer des réponses en format JSON.

Étape 1 : Créer un Serveur HTTP Basique

Le premier objectif est de créer un serveur qui écoute les requêtes HTTP et affiche une page d'accueil avec un message de bienvenue.

Tâches :

1. Créez un fichier `server.js` dans votre projet Node.js.
2. Utilisez le module `http` pour créer un serveur HTTP qui écoute sur le port 3000.
3. Le serveur doit répondre avec un message HTML indiquant "Bienvenue à Magica!" pour les requêtes GET sur la route `/`.
4. Pour toute autre URL, le serveur doit renvoyer un message d'erreur 404 indiquant "Page non trouvée".

Étape 2 : Ajouter une Page "À Propos"

Votre serveur doit désormais gérer une nouvelle route : `/about`, qui affichera une page avec un message à propos de Magica.

Tâches :

1. Ajoutez une nouvelle route `/about` au serveur qui répond avec un message HTML.
2. Le serveur doit renvoyer "À propos de Magica" pour les requêtes GET à cette URL.
3. Assurez-vous que le serveur continue à renvoyer une erreur 404 pour les autres URL non définies.

Étape 3 : Traiter une Requête POST pour Soumettre des Informations

Ajoutez la possibilité pour les utilisateurs de soumettre leur nom via une requête POST à l'URL `/submit`. Le serveur doit renvoyer un message de remerciement personnalisé contenant le nom soumis.

Tâches :

1. Ajoutez une route `/submit` pour les requêtes POST.
2. Le serveur doit lire les données envoyées par le client (le nom de l'utilisateur) et répondre avec un message du type : "Merci de nous avoir contactés, [nom]".
3. Si une requête POST est envoyée sans données, le serveur doit renvoyer un message d'erreur.

Étape 4 : Envoi de JSON en Réponse

Pour rendre votre serveur plus interactif, ajoutez une route `/api` qui renvoie une réponse JSON avec un message de bienvenue et la date actuelle.

Tâches :

1. Créez une route `/api` qui répond aux requêtes GET avec un objet JSON.

2. Cet objet doit contenir un message ("Bienvenue à l'API de Magica") et un champ indiquant la date actuelle.
 3. Assurez-vous que la réponse a le bon type de contenu (`application/json`).
-

Étape 5 (Facultative) : Ajout d'une Redirection

Ajoutez une redirection pour que les requêtes à l'URL `/old-page` soient redirigées vers `/new-page` .

Tâches :

1. Créez une route `/old-page` qui redirige de manière permanente (code HTTP 301) vers `/new-page` .
 2. Sur `/new-page` , affichez un message de bienvenue indiquant que l'utilisateur est sur la nouvelle page.
-