Collège de Bois-de-Boulogne

Journal d’apprentissage

Travail présenté à

Monsieur Mohammed Salim Meflah

Dans le cadre du cours

Système de base de données

Par:

Ebtisam ben younes

18 mai 2023

Qu’avez-vous appris sur la programmation d’arrière-plan à l’aide de Node.JS et des bases de données (5 éléments)

1. Performances élevées : Node.js est basé sur une architecture orientée événements, ce qui permet de réaliser des opérations de manière asynchrone. Cela signifie que le serveur peut traiter plusieurs requêtes simultanément, améliorant les performances globales de l'application.

2. Évolutivité : Node.js utilise un modèle de programmation non bloquant, ce qui le rend idéal pour les applications nécessitant une grande évolutivité.

3. Accès aux bases de données : Node.js dispose de bibliothèques et de packages qui permettent une intégration facile avec les bases de données.

4. Communication en temps réel : Node.js est très utilisé pour les applications nécessitant une communication bidirectionnelle en temps réel, telles que les applications de chat ou de suivi en temps réel.

5. Utilisation de requêtes API : Node.js facilite également l'utilisation et la consommation de données provenant d'API externes. Grâce à des packages tels que "Request", on peut effectuer des requêtes HTTP vers des API, collecter les données et les stocker dans une base de données pour une utilisation ultérieure.

Expliquer en vos mots, comment l’information saisie par un utilisateur dans une page web se retrouve insérée/modifiée/effacé ou interrogé dans une base de données sous une architecture web avec un serveur API REST.

Lorsqu’on utilise un serveur API REST dans une architecture web, les utilisateurs peuvent insérer, modifier, supprimer ou interroger des données dans une base de données. Voici comment cela fonctionne :

1. Insérer des données : Les utilisateurs saisissent des informations dans un formulaire ou via une requête depuis une page web. Ces informations sont envoyées au serveur API REST en utilisant une requête POST. Le serveur reçoit ces informations et les enregistre dans la base de données.
2. Modifier des données : Si les utilisateurs veulent changer des données existantes, ils envoient une requête PUT ou PATCH au serveur API REST avec les nouvelles informations. Le serveur met ensuite à jour les données correspondantes dans la base de données.
3. Supprimer des données : Si les utilisateurs veulent effacer des données, ils envoient une requête DELETE au serveur avec l’identifiant des données à supprimer. Le serveur supprime ces données de la base de données.
4. Interroger des données : Si les utilisateurs veulent récupérer des données, ils envoient une requête GET au serveur avec des critères de recherche. Le serveur trouve les données correspondantes dans la base de données et les renvoie en réponse à la requête GET.

Si vous aviez à refaire ce projet que feriez-vous différemment ?

Je pense que si j’avais à refaire ce projet j’aurai mieux organiser mon temps et mon projet vu que au début on voulait faire une site qui donne beaucoup d’information touristique sur un pays quand le nom de pays est saisi

Par contre, à cause de manque de temps on a trop simplifié notre projet pour pouvoir le remettre à temps. Donc, je pense que j’aurais investi plus de temps à l’améliorer.