Cours académique simplifié – API REST Node.js, Express et MongoDB

Introduction

Ce document présente de manière académique, simple et structurée, les notions essentielles nécessaires à la création d'une API REST complète en Node.js avec Express et MongoDB. L'objectif est de comprendre clairement les concepts fondamentaux et leur mise en œuvre.

1. Node.js

Node.js est un environnement d'exécution JavaScript orienté vers la création d'applications côté serveur. Il permet d'exécuter du JavaScript directement sur la machine de l'utilisateur, sans navigateur. Il repose sur un modèle non-bloquant et événementiel, qui le rend efficace pour les applications réseau.

2. Express

Express est un framework minimaliste construit au-dessus de Node.js. Il simplifie la création de serveurs HTTP et introduit :

- les routes (URL + méthode HTTP),

- les middlewares (fonctions exécutées avant la réponse),

- la gestion du JSON et des requêtes.

3. Concept d'API REST

Une API REST définit un ensemble de règles pour permettre à un client (navigateur, outil ou application) de communiquer avec un serveur. Les principales opérations sont :

- GET : lire une ressource

- POST : créer une ressource

- PUT : modifier une ressource

- DELETE : supprimer une ressource

4. MongoDB

MongoDB est une base de données NoSQL orientée documents. Les données y sont stockées sous forme de documents JSON. Elle ne nécessite pas de schéma strict, ce qui la rend flexible et adaptée aux APIs modernes.

5. Mongoose

Mongoose est une couche d'abstraction permettant de définir des schémas pour MongoDB et de valider les données. Il simplifie l'accès à la base en offrant :

- des modèles,

- des méthodes CRUD,

- la validation intégrée.

6. Architecture MVC

Le modèle MVC (Modèle – Vue – Contrôleur) permet d'organiser un projet backend en trois parties distinctes :

- Modèle : structure de la donnée et communication avec la base (Mongoose).

- Contrôleur : logique métier (CRUD).

- Routes : liaison entre URL et fonctions du contrôleur.

Dans ce TP, la "Vue" n'est pas utilisée, car il s'agit uniquement d'une API.

7. Mise en œuvre du TP

Étape 1 : Initialisation du projet

Lancer : npm init -y et installer express, mongoose, cors, dotenv, nodemon.

Étape 2 : Arborescence MVC

Création des dossiers models, controllers, routes et du fichier server.js.

Étape 3 : server.js

Le fichier configure express, active les middlewares, se connecte à MongoDB et charge les routes. Il constitue le point d'entrée de l'application.

Étape 4 : Modèle Product

Un schéma Mongoose définit la structure du produit : name, description, price, quantity, createdAt. Ce modèle garantit la cohérence des données.

Étape 5 : Contrôleur Product

Les cinq fonctions du CRUD remplissent les opérations REST :

- createProduct : ajoute un document

- getAllProducts : retourne tous les documents

- getProductById : retourne un document précis

- updateProduct : met à jour un document

- deleteProduct : supprime un document

Chaque fonction utilise async/await et gère les erreurs via try/catch.

Étape 6 : Routes

Le fichier de routes relie chaque URL à la fonction du contrôleur correspondante.

8. Schéma de fonctionnement

Client → Route → Contrôleur → Modèle → MongoDB

MongoDB → Modèle → Contrôleur → Route → Client

Conclusion

Ce TP permet de comprendre les bases de la construction d'une API REST moderne en Node.js. En appliquant l'architecture MVC, l'organisation du code devient claire et maintenable. L'intégration de MongoDB via Mongoose modernise la gestion de la donnée et simplifie les opérations CRUD.