

Introduction à MongoDB

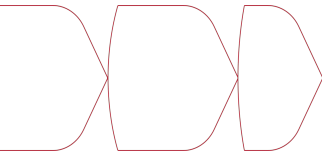


Comprendre les bases de la base de données NoSQL orientée documents

Slimani Mohamed Amine

EHTP

February 11, 2025



Sommaire

Qu'est-ce que MongoDB ?

Pourquoi utiliser MongoDB ?

Concepts de base de MongoDB

Architecture de MongoDB

Bonnes pratiques

Outils pour travailler avec MongoDB

Exemple d'application avec MongoDB et Node.js

Défis de MongoDB

Pourquoi c'est important ?

Qu'est-ce que MongoDB ?

- ▶ **Définition** : MongoDB est une base de données NoSQL orientée documents, qui stocke les données sous forme de documents JSON-like (BSON).
- ▶ **Objectif** : Offrir une solution de stockage flexible et scalable pour les applications modernes.
- ▶ **Avantages** : Flexibilité, scalabilité, performances élevées.

Pourquoi utiliser MongoDB ?

- ▶ **Flexibilité** : Pas de schéma fixe, adapté aux données non structurées ou semi-structurées.
- ▶ **Scalabilité** : Facile à scaler horizontalement avec le sharding.
- ▶ **Performances** : Accès rapide aux données grâce à l'indexation.

Concepts de base de MongoDB

- ▶ **Document** : Unité de base de données, stockée en BSON (Binary JSON).
- ▶ **Collection** : Groupe de documents, similaire à une table dans les bases de données relationnelles.
- ▶ **Base de données** : Conteneur pour les collections.

Architecture de MongoDB

- ▶ **Moteur de stockage** : WiredTiger (par défaut) pour la gestion des données.
- ▶ **Sharding** : Répartition des données sur plusieurs serveurs pour la scalabilité.
- ▶ **Réplication** : Copie des données sur plusieurs serveurs pour la haute disponibilité.

Bonnes pratiques

- ▶ **Indexation** : Créer des index pour accélérer les requêtes.
- ▶ **Schéma flexible** : Profiter de la flexibilité tout en gardant une structure logique.
- ▶ **Sécurité** : Configurer l'authentification et le chiffrement.

Outils pour travailler avec MongoDB

- ▶ **MongoDB Compass** : Interface graphique pour interagir avec MongoDB.
- ▶ **MongoDB Atlas** : Service cloud pour déployer et gérer MongoDB.
- ▶ **Mongoose** : Bibliothèque ODM (Object Data Modeling) pour Node.js.

Exemple d'application avec MongoDB et Node.js

Connexion à MongoDB avec Node.js

```
const { MongoClient } = require('mongodb');
const uri = 'mongodb://localhost:27017';
const client = new MongoClient(uri);

Comment Code
async function run() {
  try {
    await client.connect();
    const database = client.db('mydb');
    const collection = database.collection('users');
    const result = await collection.insertOne({ name: 'Alice', age: 25 });
    console.log(`Document inséré avec l'ID : ${result.insertedId}`);
  } finally {
    await client.close();
  }
}
run().catch(console.dir);
```

Défis de MongoDB

- ▶ **Gestion des schémas** : Bien que flexible, un schéma mal conçu peut entraîner des problèmes de performance.
- ▶ **Consommation de mémoire** : MongoDB peut utiliser beaucoup de mémoire pour les grandes bases de données.
- ▶ **Sécurité** : Configuration correcte nécessaire pour éviter les vulnérabilités.

Pourquoi c'est important ?

- ▶ MongoDB est une base de données puissante pour les applications modernes.
- ▶ Elle offre flexibilité, scalabilité, et performances élevées.
- ▶ Comprendre MongoDB est essentiel pour les développeurs et les administrateurs de bases de données.

MongoDB est une base de données NoSQL révolutionnaire pour le stockage flexible et scalable des données. Explorez, apprenez, et innovez avec MongoDB !