Using Mongoose with MongoDB in Node.js

Using Mongoose with MongoDB in Node.js involves several key steps to set up and interact with your database.

1. Project Setup and Installation:

Initialize a Node.js project: `npm init -y`

Install Mongoose: `npm install mongoose`

2. Connecting to MongoDB:

Require Mongoose in your application.

Use `mongoose.connect()` to establish a connection, providing your MongoDB connection URI (e.g., `mongodb://localhost:27017/yourDatabaseName` for local or a connection string from MongoDB Atlas).

Handle connection success and error events.

JavaScript

```
const mongoose = require('mongoose');

const connectDB = async () => {
  try {
    await mongoose.connect('mongodb://localhost:27017/yourDatabaseName', {
      useNewUrlParser: true,
      useUnifiedTopology: true, // Recommended for newer versions
    });
    console.log('MongoDB Connected...');
  } catch (err) {
    console.error(err.message);
    process.exit(1); // Exit process with failure
  }
};
```

```
module.exports = connectDB;
```

3. Defining Schemas and Models:

Create a Mongoose Schema to define the structure and types of your data.

Create a Mongoose Model from the Schema, which represents a collection in MongoDB and provides an interface for interacting with documents.

JavaScript

```
const mongoose = require('mongoose');

const userSchema = new mongoose.Schema({
  name: {
    type: String,
    required: true,
  },
  email: {
    type: String,
    required: true,
    unique: true,
  },
  age: Number,
});

const User = mongoose.model('User', userSchema);
module.exports = User;
```

4. Performing CRUD Operations:

Use `Model.create()` or `new Model().save()` to insert new documents.

Use `Model.find()` to retrieve multiple documents, `Model.findOne()` to retrieve a single document, or `Model.findById()` to find by ID.

Use `Model.updateOne()`, `Model.updateMany()`, or `Model.findByIdAndUpdate()` to modify existing documents.

Use `Model.deleteOne()`, `Model.deleteMany()`, or `Model.findByIdAndDelete()` to remove documents.

JavaScript

```
const User = require('./models/User'); // Assuming User model is in ./models/User.js

// Example: Creating a new user
const newUser = await User.create({ name: 'John Doe', email: 'john@example.com', age: 30
});

// Example: Finding all users
const users = await User.find();

// Example: Updating a user
await User.updateOne({ email: 'john@example.com' }, { age: 31 });

// Example: Deleting a user
await User.deleteOne({ email: 'john@example.com' });
```

5. Disconnecting from MongoDB (Optional but Recommended):

Close the MongoDB connection when your application is shutting down gracefully.

JavaScript

```
mongoose.connection.close(() => {
  console.log('Disconnected from MongoDB');
});
```

https://www.topcoder.com/thrive/articles/how-to-connect-mongodb-to-node-js-using-mongoose

Revision #3
Created 29 October 2025 02:43:36 by AI API

Updated 19 November 2025 05:24:24 by AI Channel