1. Create folder using terminal command:  
   mkdir node-express-mongo-crud  
   cd node-express-mongo-crud
2. npm init -y
3. npm install express mongoose body-parser
4. inside project folder, create server.js file and paste:

const express = require('express');

const mongoose = require('mongoose');

const bodyParser = require('body-parser');

const Item = require('./models/Item'); // Import the model

const app = express();

// Middleware

app.use(bodyParser.json());

// MongoDB connection

mongoose.connect('mongodb://localhost:27017/crud\_api', {

useNewUrlParser: true,

useUnifiedTopology: true,

})

.then(() => console.log('Connected to MongoDB'))

.catch((err) => console.error('MongoDB connection error:', err));

// CREATE - Add new item

app.post('/api/items', async (req, res) => {

const { name, description, price } = req.body;

const newItem = new Item({ name, description, price });

try {

const savedItem = await newItem.save();

res.status(201).json(savedItem);

} catch (err) {

res.status(400).json({ message: err.message });

}

});

// READ - Get all items

app.get('/api/items', async (req, res) => {

try {

const items = await Item.find();

res.status(200).json(items);

} catch (err) {

res.status(500).json({ message: err.message });

}

});

// READ - Get item by ID

app.get('/api/items/:id', async (req, res) => {

const { id } = req.params;

try {

const item = await Item.findById(id);

if (!item) return res.status(404).json({ message: 'Item not found' });

res.status(200).json(item);

} catch (err) {

res.status(500).json({ message: err.message });

}

});

// UPDATE - Update item by ID

app.put('/api/items/:id', async (req, res) => {

const { id } = req.params;

const { name, description, price } = req.body;

try {

const updatedItem = await Item.findByIdAndUpdate(

id,

{ name, description, price },

{ new: true }

);

if (!updatedItem) return res.status(404).json({ message: 'Item not found' });

res.status(200).json(updatedItem);

} catch (err) {

res.status(400).json({ message: err.message });

}

});

// DELETE - Delete item by ID

app.delete('/api/items/:id', async (req, res) => {

const { id } = req.params;

try {

const deletedItem = await Item.findByIdAndDelete(id);

if (!deletedItem) return res.status(404).json({ message: 'Item not found' });

res.status(200).json({ message: 'Item deleted' });

} catch (err) {

res.status(500).json({ message: err.message });

}

});

// Start server

const PORT = process.env.PORT || 5000;

app.listen(PORT, () => {

console.log(`Server running on port ${PORT}`);

});

--------------------------------------------------------------------------------------------------------

1. create new folder in the project folder/ models using command in the terminal:  
   mkdir models
2. inside models folder create Item.js file and paste:

const mongoose = require('mongoose');

// Define Item schema

const itemSchema = new mongoose.Schema({

  name: {

    type: String,

    required: true,

  },

  description: {

    type: String,

    required: true,

  },

  price: {

    type: Number,

    required: true,

  },

});

// Create and export Item model

module.exports = mongoose.model('Item', itemSchema);

1. Open POSTMAN app:  
   a) click on collection  
   b) create new collection  
   c) give any name to it  
   d) New request  
   e) create item POST and paste:  
   {  
    "name": "OPPO",  
    "description": "Fast and secure phone company",  
    "price": 20000  
   }  
     
   id will get created after clicking SEND  
     
   f) create item GET:  
   <http://localhost:5000/api/items>  
   g) GET [http://localhost:5000/api/items/<ITEM\_ID](http://localhost:5000/api/items/%3cITEM_ID)>  
   h) PUT [http://localhost:5000/api/items/<ITEM\_ID](http://localhost:5000/api/items/%3cITEM_ID)>  
   i) DELETE [http://localhost:5000/api/items/<ITEM\_ID](http://localhost:5000/api/items/%3cITEM_ID)>