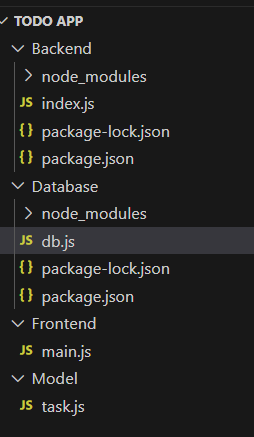
Node/express



task.js

// models/Task.js

const mongoose = require("mongoose");

const TaskSchema = new mongoose.Schema({

  id: Number,

  taskName: String

});

module.exports = mongoose.model("Task", TaskSchema);

main.js

function add(task) {

  return new Promise((resolve, reject) => {

    fetch("http://localhost:3000/api/tasks/add", {

      method: "POST",

      headers: {

        "Content-Type": "application/json"

      },

      body: JSON.stringify(task)  // Send the task directly

    })

    .then(res => res.json())

    .then(data => {

      if (data.status == "success") {

        resolve(data.data);

      } else {

        reject(data.message);

      }

    })

    .catch(err => {

      reject(err.message);

    });

  });

}

function getAllTasks() {

  return new Promise((resolve, reject) => {

    fetch("http://localhost:3000/api/tasks/getAll", {

      method: "GET",

      headers: {

        "Content-Type": "application/json"

      }

    })

    .then(res => res.json())

    .then(data => {

      if (data.status === "success") {

        resolve(data.data.taskArray);  // Corrected to match backend structure

      } else {

        reject(data.message);

      }

    })

    .catch(err => {

      reject(err.message);

    });

  });

}

function update(task) {

  return new Promise((resolve, reject) => {

    fetch("http://localhost:3000/api/tasks/update", {

      method: "PUT",

      headers: {

        "Content-Type": "application/json"

      },

      body: JSON.stringify(task)  // Send the task directly

    })

    .then(res => res.json())

    .then(data => {

      if (data.status == "success") {

        resolve(data.data);

      } else {

        reject(data.message);

      }

    })

    .catch(err => {

      reject(err.message);

    });

  });

}

function deleteTask(task) {

  return new Promise((resolve, reject) => {

    fetch("http://localhost:3000/api/tasks/delete", {

      method: "DELETE",

      headers: {

        "Content-Type": "application/json"

      },

      body: JSON.stringify(task)  // Send the task directly

    })

    .then(res => res.json())

    .then(data => {

      if (data.status == "success") {

        resolve(data.data);

      } else {

        reject(data.message);

      }

    })

    .catch(err => {

      reject(err.message);

    });

  });

}

async function main() {

  // Add tasks

  await add({ id: 1, taskName: "complete express.js" });

  await add({ id: 2, taskName: "complete node.js" });

  await add({ id: 3, taskName: "master javascript" });

  await add({ id: 4, taskName: "play with task 1, 2, 3" });

  await add({ id: 5, taskName: "learn new techstack"});

  // Fetch all tasks

  let g = await getAllTasks();

  console.log(g);  // Should print all tasks

  await update({ id: 1, taskName: "complete and nail express.js" });

  console.log( await getAllTasks());

  await deleteTask({ id: 5, taskName: "learn new techstack"});

  console.log( await getAllTasks());

}

main();

db.js

const mongoose=require("mongoose");

const connectDB=async()=>{

    try{

        await mongoose.connect("mongodb://localhost:27017/todo",{

            useNewUrlParser:true,

            useUnifiedTopology:true

        });

        console.log("MongoDB connected successfully");

      }

      catch(err){

        console.error("MongoDB connection failed",err);

      }

};

// connectDB();

module.exports=connectDB;

index.js

const express = require('express');

const connectDB = require('../Database/db');  // ✅ correct

const Task = require('../Model/task');        // ✅ correct

const app = express();

const port = 3000;

// Connect to DB

connectDB();

// Middleware to parse JSON

app.use(express.json());

// Add task

app.post("/api/tasks/add", async (req, res) => {

  try {

    const { id, taskName } = req.body;

    if (!taskName) {

      return res.status(400).json({ status: "error", message: "Task is required" });

    }

    const newTask = new Task({ id, taskName });

    await newTask.save();

    res.json({ status: "success", data: "Task added" });

  } catch (err) {

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

  }

});

// Get all tasks

app.get("/api/tasks/getAll", async (req, res) => {

  try {

    const tasks = await Task.find();

    if (tasks.length === 0) {

      return res.status(404).json({ status: "error", message: "No tasks found" });

    }

    res.json({ status: "success", data: { taskArray: tasks } });

  } catch (err) {

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

  }

});

// Update task

app.put("/api/tasks/update", async (req, res) => {

  try {

    const { id, taskName } = req.body;

    if (!taskName) {

      return res.status(400).json({ status: "error", message: "Task is required" });

    }

    const updated = await Task.findOneAndUpdate({ id }, { taskName }, { new: true });

    if (!updated) {

      return res.status(404).json({ status: "error", message: "Task not found" });

    }

    res.json({ status: "success", data: "Task updated" });

  } catch (err) {

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

  }

});

// Delete task

app.delete("/api/tasks/delete", async (req, res) => {

  try {

    const { id } = req.body;

    const result = await Task.deleteOne({ id });

    if (result.deletedCount === 0) {

      return res.status(404).json({ status: "error", message: "Task not found" });

    }

    res.json({ status: "success", data: "Task deleted" });

  } catch (err) {

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

  }

});

// Start server

app.listen(port, () => {

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

});

Without databse

// Add task

function add(task) {

  return fetch("http://localhost:3000/api/tasks/add", {

    method: "POST",

    headers: { "Content-Type": "application/json" },

    body: JSON.stringify(task)

  })

  .then(res => res.json());

}

// Get all tasks

function getAllTasks() {

  return fetch("http://localhost:3000/api/tasks/getAll")

    .then(res => res.json())

    .then(data => data.data.taskArray);

}

// Update task

function updateTask(task) {

  return fetch("http://localhost:3000/api/tasks/update", {

    method: "PUT",

    headers: { "Content-Type": "application/json" },

    body: JSON.stringify(task)

  })

  .then(res => res.json());

}

// Delete task

function deleteTask(task) {

  return fetch("http://localhost:3000/api/tasks/delete", {

    method: "DELETE",

    headers: { "Content-Type": "application/json" },

    body: JSON.stringify(task)

  })

  .then(res => res.json());

}

// Main function

async function main() {

  await add({ id: 1, taskName: "complete express.js" });

  await add({ id: 2, taskName: "complete node.js" });

  await add({ id: 3, taskName: "master javascript" });

  await add({ id: 4, taskName: "play with 1, 2, 3" });

  await add({ id: 5, taskName: "Learn a new techstack" });

  // console.log("After Adding:");

  console.log(await getAllTasks());

  await updateTask({ id: 5, taskName: "Build full stack TODO app" });

  console.log("After Updating:");

  console.log(await getAllTasks());

  await deleteTask({ id: 5});

  console.log("After Deletion:");

  console.log(await getAllTasks());

}

main();

// Add task

function add(task) {

  return fetch("http://localhost:3000/api/tasks/add", {

    method: "POST",

    headers: { "Content-Type": "application/json" },

    body: JSON.stringify(task)

  })

  .then(res => res.json());

}

// Get all tasks

function getAllTasks() {

  return fetch("http://localhost:3000/api/tasks/getAll")

    .then(res => res.json())

    .then(data => data.data.taskArray);

}

// Update task

function updateTask(task) {

  return fetch("http://localhost:3000/api/tasks/update", {

    method: "PUT",

    headers: { "Content-Type": "application/json" },

    body: JSON.stringify(task)

  })

  .then(res => res.json());

}

// Delete task

function deleteTask(task) {

  return fetch("http://localhost:3000/api/tasks/delete", {

    method: "DELETE",

    headers: { "Content-Type": "application/json" },

    body: JSON.stringify(task)

  })

  .then(res => res.json());

}

// Main function

async function main() {

  await add({ id: 1, taskName: "complete express.js" });

  await add({ id: 2, taskName: "complete node.js" });

  await add({ id: 3, taskName: "master javascript" });

  await add({ id: 4, taskName: "play with 1, 2, 3" });

  await add({ id: 5, taskName: "Learn a new techstack" });

  // console.log("After Adding:");

  console.log(await getAllTasks());

  await updateTask({ id: 5, taskName: "Build full stack TODO app" });

  console.log("After Updating:");

  console.log(await getAllTasks());

  await deleteTask({ id: 5});

  console.log("After Deletion:");

  console.log(await getAllTasks());

}

main();

======================================================================