Code :

```
const express = require('express');
const app = express();
app.use(express.json());

// Store tasks in memory
let tasks = [];

// GET all tasks
app.get('/tasks', (req, res) => {
  res.status(200).json(tasks);
});

// GET a specific task by ID
app.get('/tasks/:id', (req, res) => {
  const taskId = req.params.id;
  const task = tasks.find(task => task.id === taskId);

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

  res.status(200).json(task);
});

// POST create a new task
app.post('/tasks', (req, res) => {
  const { title, description } = req.body;

  if (!title || !description) {
    return res.status(400).json({ message: 'Title and description are required' });
  }

  const newTask = { id: tasks.length + 1, title, description };
  tasks.push(newTask);

  res.status(201).json(newTask);
});

// PUT update an existing task by ID
app.put('/tasks/:id', (req, res) => {
  const taskId = req.params.id;
  const { title, description } = req.body;
  const taskIndex = tasks.findIndex(task => task.id === taskId);

  if (taskIndex === -1) {
    return res.status(404).json({ message: 'Task not found' });
  }

  if (!title || !description) {
    return res.status(400).json({ message: 'Title and description are required' });
  }

  tasks[taskIndex] = { ...tasks[taskIndex], title, description };

  res.status(200).json(tasks[taskIndex]);
});

// DELETE a task by ID
app.delete('/tasks/:id', (req, res) => {
```

```
  const taskId = req.params.id;
  const taskIndex = tasks.findIndex(task => task.id === taskId);

  if (taskIndex === -1) {
    return res.status(404).json({ message: 'Task not found' });
  }

  tasks.splice(taskIndex, 1);

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

// Error handling middleware
app.use((err, req, res, next) => {
  console.error(err.stack);
  res.status(500).json({ message: 'Something went wrong!' });
});

// Start the server
const PORT = process.env.PORT || 3000;
app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});
```