

Lab: Build a Todo Application with Express.js

Objective:

In this lab, you will create a simple RESTful API using Express.js to manage a Todo list. The API will allow users to:

- Create a new todo (default status: "in-progress")
- List all todos
- Get a single todo by ID
- Update a todo
- Delete a todo

Requirements:

Each todo should have the following structure:

```
{
  "id": 1,
  "title": "Learn Express.js",
  "status": "in-progress"
}
```

- 1.
2. Store todos in a file named `db.json` (use JSON format).
3. Implement the following API endpoints:
 - `POST /todos` - Create a new todo (default status: "in-progress")
 - `GET /todos` - List all todos
 - `GET /todos/:id` - Get a single todo by ID
 - `PUT /todos/:id` - Update a todo (modify title or status)
 - `DELETE /todos/:id` - Delete a todo

Boilerplate Code:

Here's a starting point for your project:

```
const express = require('express');
const fs = require('fs');
const app = express();
const port = 4000;

app.use(express.json());

// Function to read todos from file
const readTodos = () => {
  const data = fs.readFileSync('db.json', { encoding: 'utf-8' });
  return JSON.parse(data);
};
```

```

// Function to write todos to file
const writeTodos = (todos) => {
  fs.writeFileSync('db.json', JSON.stringify(todos, null, 2));
};

// Create a new todo
app.post('/todos', (req, res) => {
  });

// List all todos
app.get('/todos', (req, res) => {
  res.json(readTodos());
});

// Get a single todo by ID
app.get('/todos/:id', (req, res) => {
  });

// Update a todo
app.put('/todos/:id', (req, res) => {
  });

// Delete a todo
app.delete('/todos/:id', (req, res) => {
  });

app.listen(port, () => {
  console.log(`Server running on port ${port}`);
});

```

Instructions:

1. Setup your environment:

- Install Node.js and Express if not installed.
- Create a new directory for your project.
- Run `npm init -y` to create a `package.json` file.
- Run `npm install express` to install Express.js.
- Create `server.js` and `db.json` (initialize it with `[]`).

2. Implement the missing parts:

- Make sure all API endpoints work as expected.
- Test using Postman, Thunder Client, or `curl`.

3. Bonus Tasks:

- Add error handling for invalid JSON inputs.
- Implement a feature to mark a todo as "completed".
- Ensure unique IDs for todos.

Submission:

Once completed, submit your project with a README explaining how to run the API and test the endpoints.