MVC in Node.js mit Express

Einleitung

Dieses Projekt zeigt eine einfache Umsetzung des **MVC-Patterns** (Model-View-Controller) mit **Node.js** und **Express**.

Das MVC-Pattern hilft dabei, Code in drei Hauptbereiche zu unterteilen:

- **Model (Daten & Logik)** – Verarbeitet und speichert Daten.
- **View (Benutzeroberfläche)** – Stellt Daten dar und interagiert mit dem Benutzer.
- **Controller (Steuerung)** – Vermittelt zwischen Model und View.

Projektstruktur

```
mvc-example/
├── controllers/
│   ├── userController.js   # Logik zur Verarbeitung von Benutzeranfragen
├── models/
│   ├── userModel.js        # Speichert und verwaltet Benutzerdaten
├── views/
│   ├── index.ejs          # HTML-Template für die Darstellung der Daten
├── routes/
│   ├── userRoutes.js       # Definiert die API-Endpunkte
├── public/
│   ├── style.css          # Statische Dateien (CSS, Bilder etc.)
├── app.js                 # Hauptdatei der Anwendung
├── package.json           # Projekt- und Abhängigkeitsverwaltung
```

Bestandteile des MVC

1. Model (`models/userModel.js`)

Das Model verwaltet die Daten und enthält die Geschäftslogik.

```
const users = [{ id: 1, name: "Alice" }, { id: 2, name: "Bob" }];

function getAllUsers() {
    return users;
}

function addUser(name) {
    const newUser = { id: users.length + 1, name };
    users.push(newUser);
    return newUser;
}

module.exports = { getAllUsers, addUser };
```

2. Controller (`controllers/userController.js`)

Der Controller verarbeitet Anfragen und steuert die Datenverarbeitung.

```
const UserModel = require("../models/userModel");
```

```
function getUsers(req, res) {
    const users = UserModel.getAllUsers();
    res.render("index", { title: "User List", users });
}

function addUser(req, res) {
    const name = req.body.name;
    if (name) {
        UserModel.addUser(name);
    }
    res.redirect("/users");
}

module.exports = { getUsers, addUser };
```

3. Routes (`routes/userRoutes.js`)

Hier werden die Endpunkte definiert, über die der Controller aufgerufen wird.

```
const express = require("express");
const router = express.Router();
const userController = require("../controllers/userController");

router.get("/", userController.getUsers);
router.post("/add", userController.addUser);

module.exports = router;
```

4. View (`views/index.ejs`)

Die View zeigt die Benutzerliste und ermöglicht das Hinzufügen von Benutzern.

```
<!DOCTYPE html>
<html lang="de">
<head>
    <meta charset="UTF-8">
    <title><%= title %></title>
    <link rel="stylesheet" href="/style.css">
</head>
<body>
    <h1>User List</h1>
    <ul>
        <% users.forEach(user => { %>
            <li><%= user.name %></li>
        <% }); %>
    </ul>
    <form action="/users/add" method="POST">
        <input type="text" name="name" placeholder="Neuer Benutzer">
        <button type="submit">Hinzufügen</button>
    </form>
</body>
</html>
```

Installation & Start

1. **Abhängigkeiten installieren:**

   ```
   npm install express ejs
   ```

2. **Server starten:**

```
node app.js
```

3. **Im Browser öffnen:** `http://localhost:3000/users`

Aufgabe: Benutzerverwaltung mit Lösch- und Bearbeitungsfunktion

Aufgabe:
Erweitere das bestehende MVC-Projekt, um folgende Funktionen hinzuzufügen:

1. **Benutzer löschen:** Ein Button neben jedem Benutzer soll es ermöglichen, ihn zu löschen.
2. **Benutzer bearbeiten:** Ein Benutzername soll bearbeitet und aktualisiert werden können.