LU03.A04 - Webformulare: JSON-Daten lesen, bearbeiten, einfügen und löschen

1. Bestandteile eine node-Servers kennen.
2. Basis-Server ausführen können.
3. Daten anzeigen, verändern und schreiben können.

Rahmenbedingungen

- Sozialform: individual
- Hilfsmittel: OpenBooks
- Zeit: 30 Minutes
- Erwartetes Resultat: Node-Server, der verschiedene abgefragte Daten am Monitor/Browser bearbeitet und ausgibt.

Ausgangslage

...

Ordnerstruktur der Daten

Um Rahmen des Web-Entwicklungsprojekts „Personenverzeichnis“ soll eine bestehende Node.js-Applikation mit Express und EJS (Embedded JavaScript) finalisiert und in Betrieb genommen werden. Bisher blockieren fehlerhafte Dateipfade, unvollständige Routen-Variablen sowie Caching-Probleme im Firefox-Browser das fehlerfreie Hinzufügen, Bearbeiten und Löschen von Datensätzen. Ihre Aufgabe ist es, die Ordnerstruktur zu bereinigen, die Server-Logik zu vervollständigen und die Benutzeroberfläche mit einem dynamischen Hobby-Dropdown sowie optimierten Buttons auszustatten. Der lokale Prototyp muss anschliessend auf Port 3000 fehlerfrei und stabil laufen.

Benötigte File-Struktur

```
▼ a04 (Projekt-Wurzelverzeichnis)
  ▼ public
    ▼ css
        style.css
  ▼ views
      index.ejs
    data.json
    server.js
```

```
mein-projekt/
├── data.json
├── server.js
├── public/
│   └── css/
│       └── style.css
└── views/
    └── index.ejs
```

Auftrag

Nachfolgend finden Sie verschiedene Code-Snippets.

1. Fügen Sie diese zusammen.
2. Lassen Sie diese laufen
3. Prüfen Sie Ausgabe im Monitor/Browser

1. Vorarbeiten

1. Erstellen Sie in Ihrem Visual Studio Code eine Ordner mit dem Titel *lu03_a02*
2. Gehen Sie im Terminial in diesen neuen Ordner
3. Installieren Sie alle relevanten JS-Komponenten in diesem Ordner durch die nachfolgenden zwei Codezeilen –> Es werden nun verschiedene Componenten installiert (sichtbar rechts unten im CodeEditor)

```
npm init -y
npm install express
```

2. Server

Erstellen Sie ein weiteres File mit dem Titel *server.js* und kopieren den nachfolgenden Code rein.

```
const express = require('express');
const app = express();
const PORT = 3000;

// Middleware, damit Express JSON-Daten im Body lesen kann
app.use(express.json());

// Unser "Datenbank"-Ersatz
let users = ["Alice", "Bob"];

// --- DEINE AUFGABE AB HIER ---
```

```
// 1. GET ROUTE: Gib das 'users' Array zurück
app.get('/users', (req, res) => {
    // CODE HIER
});

// 2. POST ROUTE: Füge einen neuen User hinzu
app.post('/users', (req, res) => {
    const newUser = req.body.name; // Wir erwarten { "name": "DeinName" }
    if (newUser) {
        // CODE HIER: User zum Array hinzufügen
        res.status(201).send(`User ${newUser} wurde hinzugefügt.`);
    } else {
        res.status(400).send("Fehler: Kein Name angegeben!");
    }
});

// --- ENDE DER AUFGABE ---

app.listen(PORT, () => {
    console.log(`Server läuft auf http://localhost:${PORT}`);
});
```

Lösungen

LU03.L01

Volkan Demir

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/de/modul/m307/learningunits/lu03/aufgaben/04?rev=1779800115

Last update: **2026/05/26 14:55**