# LU03.A03/L03 - Datenmanagement mit Node-Server

1. Programmierung eines Node.js Webservers, der mittels GET ein Liste von Namen mittels ausgibt.
2. Programmierung eines Node.js Webservers, der in eine Liste einen Namen mittels POST hinzufügt.

## Rahmenbedingungen

- Sozialform: individual
- Hilfsmittel: Node-Webserver als JavaScript-Code
- Zeit: 20 Minuten
- Erwartetes Resultat: Node-Server, der abgefragte Daten am Monitor/Browser ausgibt und Daten im Array ergänzen kann.

## Ausgangslage

Wir haben die Grundfunktionen eine Webservers in der letzten Aufgabe kennen gelernt. Nun wollen wir speziefischer werden und die Methoden POST und GET gezielt anwenden.

**Hinweis:** Um diesen zu Programmieren, nutzen wir hierfür das JavaScript-Framework Express, welches in der Branche Standard ist.

## Vorarbeit

1. Installieren Sie im Visual Studio Code das Framework „express“ in ihr Visual Studio Code durch die nachfolgenden zwei Codezeilen.
2. npm init -y: package.json-Datei wird erstellt, dass alle Abhängigkeiten des Projektes verwaltet.
3. npm install express: Installation des express-Frameworks mit wichtigen Funktionen/Tools des Servers

### Projekt-Abhängigkeiten verwalten

```
npm init -y
```

**Ergebnis** Das Ergebnis im Terminalfenster von Visual Studio Code sollte wie folgt aussehen.

```
PS C:\Users\volka\OneDrive - Bildungszentrum Zürichsee\--- Latest Version ---\JS-Workspace> npm init -y
Wrote to C:\Users\volka\OneDrive - Bildungszentrum Zürichsee\--- Latest Version ---\JS-Workspace\package.json:

{
  "name": "js-workspace",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "type": "commonjs"
}
```

### Framework express installieren

```
npm install express
```

**Ergebnis** Das Ergebnis im Terminalfenster von Visual Studio Code sollte wie folgt aussehen.

```
PS C:\Users\volka\OneDrive - Bildungszentrum Zürichsee\--- Latest Version ---\JS-Workspace> npm install express

added 65 packages, and audited 66 packages in 3s

22 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

# Aufgabe

Das nachfolgende Script enthält einen JSON-Satz, der am Monitor anzeigt. Damit Sie das Script zum laufen bekommen befolgen Sie die nachfolgenden Anweiseungen.

1. Kopieren Sie die nachfolgende JS-Vorlage in Ihre Visual Studio Code.
2. Starten Sie den Server
3. Lassen Sie sich die Daten im Browserfenster anzeigen.
4. Ergänzen Sie den angezeigten Person-Datensatz durch weitere mindestens 3 Personen.
5. Lassen Sie sich durch eine AI eine passende HTML-Darstellung für die Daten generieren.

**server.js„“

```
const http = require('http');

// Die Daten der drei Personen
const personen = [
    { id: 1, name: "Max Mustermann", alter: 28, beruf: "Entwickler" }
];

// Server erstellen
const server = http.createServer((req, res) => {
    // Wichtig: Wir setzen den Content-Type auf 'application/json'
```

```
    res.writeHead(200, { 'Content-Type': 'application/json' });

    // Die Personen-Objekte in einen JSON-String umwandeln und senden
    res.end(JSON.stringify(personen));
});

// Server auf Port 3000 starten
const PORT = 3000;
server.listen(PORT, () => {
    console.log(`Server läuft unter http://localhost:${PORT}`);
});
```

—-

Volkan Demir

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

Permanent link:
**https://wiki.bzz.ch/de/modul/m307/learningunits/lu03/aufgaben/03?rev=1779196359**

Last update: **2026/05/19 15:12**