

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: 30 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

1. Kopieren Sie die nachfolgende JS-Vorlage in Ihre Visual Studio Code (Filename lu03\_a02.js)
2. Starten Sie den Server
3. Fragen Sie die Liste der Namen ab durch z.B. die URL: <http://localhost:3000/users>
4. Fügen Sie in die Liste Namen hinzu durch z.B. die URL: <http://localhost:3000/add/Lukas>
5. Prüfen Sie durch erneute Abfrage der List, ob Sie erfolgreich einen Namen anhängen konnten.

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

// Middleware, um JSON-Daten verarbeiten zu können
app.use(express.json());

// Unser Datenspeicher im Arbeitsspeicher
let benutzerListe = ["Maximilian", "Sophie"];

// 1. Aufgabe: Die GET-Route vervollständigen
app.get('/users', (req, res) => {
  // Hier senden wir das gesamte Array als JSON-Antwort an den Client
  zurück
  res.json(benutzerListe);
});
```

```
});

// 2. Aufgabe: Die POST-Route vervollständigen
app.post('/users', (req, res) => {
  // Wir extrahieren den Namen aus dem Request-Body
  const neuerName = req.body.name;

  if (neuerName) {
    // Der Name wird mit .push() am Ende des Arrays hinzugefügt
    benutzerListe.push(neuerName);

    // Wir senden eine Erfolgsmeldung mit dem Statuscode 201 (Created)
    res.status(201).send(`Benutzer ${neuerName} wurde erfolgreich
gespeichert.`);
  } else {
    // Falls kein Name mitgeschickt wurde, senden wir den Statuscode 400
    (Bad Request)
    res.status(400).send("Fehler: Es wurde kein Name übertragen.");
  }
});

app.listen(PORT, () => {
  console.log(`Ihr Server ist nun unter http://localhost:${PORT}
erreichbar.`);
});
```



Volkan Demir

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

<https://wiki.bzz.ch/de/modul/m307/learningunits/lu03/aufgaben/02?rev=1777878341>Last update: **2026/05/04 09:05**