WEB-TECHNOLOGIEN

ÜBUNG 10: NODE.JS

AUFGABE 1: SYNCHRONE UND ASYNCHRONE IO

Ergänzen Sie das unten stehende Node.js-Programm `file.js` folgendermaßen:

1. Das Programm soll die Zeichenkette "Hallo WEB1" in eine Datei `test.txt` schreiben.
2. Danach soll das Programm die Datei wieder einlesen und den gelesenen Inhalt auf der Konsole ausgeben.
3. Die Ausgabe des Programmes soll folgendermaßen aussehen (Reihenfolge der Ausgaben beachten!):

```
Datei schreiben und lesen:
Gelesener Inhalt: Hallo WEB1
```

4. Sorgen Sie dafür, dass die Dateisystemoperationen den Programmfluss nicht blockieren.

file.js

```
const fs = require("fs");

// Hier den Code einfügen!

console.log("Datei schreiben und lesen:");
```

API-Hilfe:

Datei schreiben:

- `writeFileSync(fileName, data)`
- `writeFile(fileName, data, callback)`, Parameter für `callback`: `err`

Datei lesen:

- `readFileSync(fileName)`
- `readFile(fileName, callback)`, Parameter für `callback`: `err, data`

AUFGABE 2: KLEINER WEB-SERVER

Realisieren Sie mit Hilfe des "http"-Moduls von Node.js einen Web-Server mit folgenden Eigenschaften:

1. Der Web-Server soll über die URL http://localhost:8080 erreichbar sein
2. Bei Aufruf der URL soll der Web-Server stets den Inhalt der HTML-Seite `welcome.html` liefern. Diese liegt auf dem Web-Server als Datei vor.

Hinweis:

Verwenden Sie das "fs"-Modul zum Einlesen der HTML-Datei.

AUFGABE 3: GRÖSSERER WEB-SERVER

Realisieren Sie mit Hilfe des "http"-Moduls von Node.js einen Web-Server, der für jede Anfrage eine einfache HTML-Seite mit der bekannten HTML-Grundstruktur zurückliefert. Im Body der HTML-Seite soll lediglich ein Text stehen, der abhängig von der Anfrage variiert (siehe 2.). Der Web-Server soll aus folgenden Dateien bestehen:

1. `server.js`: Über diese Datei soll der Web-Server gestartet werden können. Er soll nach dem Start über die URL http://localhost:3000 erreichbar sein.
2. `router.js`: Dieses *Modul* soll den Request-Listener beinhalten. Dieser soll eintreffende Anfragen folgendermaßen behandeln:
   - Bei GET-Anfragen auf die URL http://localhost:3000/feedback soll die gelieferte HTML-Seite den Text "Liste der Feedbacks" enthalten.
   - Bei *allen anderen* Anfragen soll die gelieferte HTML-Seite die HTTP-Methode und die URL der Anfrage als Text enthalten.

   Versehen Sie die HTTP-Antwort stets mit einem passenden HTTP-Statuscode.
3. `template.js`: Dieses *Modul* soll die HTML-Seite mit Hilfe eines Template-Literals dynamisch erzeugen.