

	Datum	14.01.2022
Informatik Web Application Engineering		Übungsblatt 12
Thomas Feilhauer		Zu lösen bis 21.01.2022, 8:00 Uhr

## Evaluation der Lehrveranstaltung am 14.01.2022

### ***Vorbereitungsaufgabe (in der Übungsstunde zu lösen):***

Installieren Sie Node.js. Erzeugen Sie ein neues Node.js-Projekt. Entwickeln Sie in Node.js einen Intermediary-HTTP-Server, der von einem (anderen, bereits lokal oder remote laufenden) HTTP-Server Daten (im XML- oder JSON-Format) anfordert und diese an den anfordernden User Agent weiterleitet. Die Response ist dabei unabhängig von der (seitens User Agent) anfordernden URL, d.h. der User Agent übergibt keine Parameter an den Node.js-HTTP-Server.

Folgende Web-Seiten können beim Lösen der Aufgabe(n) behilflich sein:

Node.js:

<https://nodejs.org/>

<https://nodejs.org/en/docs/>

<https://nodejs.dev/>

<https://code.visualstudio.com/docs/nodejs/nodejs-tutorial>

<https://www.guru99.com/node-js-tutorial.html>

[https://dev.to/abdisalan\\_js/how-to-code-a-video-streaming-server-using-nodejs-2o0](https://dev.to/abdisalan_js/how-to-code-a-video-streaming-server-using-nodejs-2o0)

### ***Aufgabe 18: Node.js Video-Stream***

***(7 Punkte)***

Entwickeln Sie eine Web-Applikation mit Node.js, die es den BenutzerInnen ermöglicht, über einen Web-Browser einen Video-Stream vom HTTP-Server anzufordern und im Web-Browser abspielen zu lassen.

Dazu sollte der User zunächst über ein Text-Eingabefeld den Namen eines Films (per Dateinamen) eingeben können, den er gerne sehen möchte. Dieser Eingabe-String soll per HTTP-Request nach Betätigen eines Submit-Buttons an den Node.js-Server geschickt werden. Der Node.js-Server soll anschließend prüfen, ob im Root-Verzeichnis des Web-Servers oder einem darin liegenden Unterverzeichnis eine Video-Datei mit dem gesuchten Namen vorhanden ist. Wird keine Video-Datei mit diesem Namen gefunden, dann soll eine angemessene Fehlermeldung an den User-Agent zurückgegeben werden. Falls eine entsprechende Datei vorhanden ist, dann soll diese per Stream an den anfordernden Web-Browser übertragen werden und dort zur Anzeige gebracht werden. Verwenden Sie dazu die von HTML5 bereitgestellten Features zur Ausgabe von Videos. Achten Sie insbesondere darauf, die HTTP-Header-Felder korrekt zu setzen:

- Accept-Ranges
- Content-Type
- Content-Length
- Content-Range

Ihre Lösung muss zumindest das MP4-Videoformat unterstützen. Testen Sie Ihre Lösung auch mit Videos, die mehrere Minuten lang sind.

Dokumentieren Sie zumindest die vollständige URL, unter der die Einstiegsseite Ihrer Webapplikation im Web-Browser aufgerufen werden kann, die unterstützten Video-Formate sowie das Root-Verzeichnis, in dem die Video-Files abgelegt werden können.