

A4 Web-Labor - Bookstore VueJS Frontend (GK+EK)

Done

Voraussetzungen

- installierte Node.js-Umgebung inkl. npm-Paketmanager: <https://nodejs.org/en/download/>
- Lösung für A3

Aufgabe

In der Aufgabe A3 haben wir einen kleinen Webserver programmiert, welcher es uns ermöglicht, Bücher zu verwalten. Nun wollen wir dafür in VueJS ein Frontend basteln!

NodeJS Server

Damit unser Webserver Anfragen von localhost entgegennimmt, müssen wir dies zuerst aktivieren, da man sonst einen CORS-Fehler bekommt. Dies geht recht einfach, indem man gesendeten Responses noch bestimmte Header hinzufügt:

```
http.createServer(function (req, res) {
  const headers = [
    "Access-Control-Allow-Origin": "*",
    "Access-Control-Allow-Methods": "OPTIONS, POST, GET",
    "Access-Control-Max-Age": 2592000,
    "Content-Type": 'text/json'
  ];
  res.writeHead(200, headers);
  ...
}
```

Nun sollte euer NodeJS-Webserver bereit sein.

VueJS Frontend

Wir erinnern uns an die Grundlagen von VueJS - als Erinnerung: siehe Aufgabe A2. Tipp: Verwende eine Kopie deiner Lösung für A2 als Vorlage.

Unser VueJS Frontend braucht zunächst eine Liste der Bücher, die es anzeigen soll:

```
var app = new Vue({
  el: '#app',
  data: {
    buecher: []
  },
  ...
})
```

In VueJS können HTML-Elemente über v-for wiederholt angezeigt werden:

```
<div v-for="b in buecher">
  {{b.name}}
</div>
```

Alles, was nun innerhalb dieses div-Elements geschrieben wird, wird für jedes Buch angezeigt.

Um aus JavaScript auf Webserver zuzugreifen, kann neben jQuery auch die Axios-Bibliothek verwendet werden. Diese importieren wir am besten wie gewohnt über die CDN im head:

```
<head>
...
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
</head>
```

Hat man Axios importiert, kann man nun jederzeit im Code Requests abschicken:

```
methods: {
  refresh () {
    axios.get('http://localhost:8080/')
      .then((response) => {
        this.buecher = response.data;
      });
  },
  ...
}
```

In diesem Beispiel wird ein GET-Request an localhost (Loopback-Adresse, d.h. auf die eigene IP-Adresse) geschickt. Das Ergebnis wird direkt in das Attribut buecher gespeichert - unser NodeJS-Server liefert ja bereits ein fertiges JSON-Array zurück.

In VueJS gibt es eine spezielle Methode namens mounted(), welche einmalig aufgerufen wird, wenn die Seite fertig geladen hat und VueJS fertig initialisiert wurde:

```
mounted() {
  this.refresh();
},
methods: {
  ...
}
```

mounted() befindet sich wohlgermerkt nicht im Methods-bereich, sondern steht auf gleicher Ebene wie data und methods.

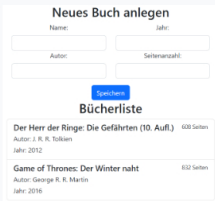
Mit axios können nicht nur GET-Requests geschickt werden - es gibt auch axios.post, axios.delete und axios.put.

Anforderungen

Entwickle ein VueJS-Frontend für deine Buchhandlung!

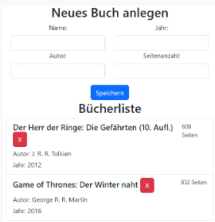
GKÜ:

- Das Frontend stellt nach dem Öffnen die Liste aller Bücher grafisch dar - diese werden vom NodeJS-Backend eingelesen
- Das Frontend erlaubt das Hinzufügen neuer Bücher über entsprechende Eingabefelder - die Eingaben werden direkt an das Backend geschickt. Nach dem Hinzufügen wird die Bücherliste automatisch aktualisiert.



GKV:

- Bücher können auch über einen Button gelöscht werden - dabei wird ein entsprechender Request an das NodeJS-Backend geschickt. Nach dem Löschen wird die Bücherliste automatisch aktualisiert.



EKÜ:

- Das Frontend verwendet GET-Requests fürs Auslesen, POST fürs Hinzufügen und DELETE zum Löschen
- Das Frontend ist visuell ansprechend und responsiv

EKV:

- Das Frontend unterstützt auch das Bearbeiten von Büchern - hierfür wird ein PUT-Request an das Backend geschickt. Nach dem Bearbeiten wird die Bücherliste automatisch aktualisiert.

Abgabe

Ladet eure Abgabe hier mit dem Namen "klasse_nachname.js" und "klasse_nachname.html" hoch und kommt zu einem **Abgabegespräch**. Es ist **kein** Protokoll nötig.

Add submission

Submission status

Submission status	No submissions have been made yet
Grading status	Graded
Last modified	-
Submission comments	▶ Comments (0)

Feedback

Grade	EK vollständig
Graded on	Wednesday, 1 March 2023, 1:04 PM
Graded by	DD Dolezal Dominik