

## A2 Web-Labor - Schülerformular und VueJS (GK+EK)

Done

### Aufgabe

In dieser Aufgabe wollen wir unser Schülerformular mit einem modernen Frontend-Framework erweitern, nämlich mit VueJS.

Bitte **kopiere das Schülerformular** aus der vorigen Aufgabe und **entferne den bisherigen JavaScript-Code** (Input-Validierung).

Wir möchten das Formular um einfache Funktionalitäten von VueJS erweitern, wodurch wir - sehr einfach - dynamische Webseiten erstellen können.

### VueJS Grundlagen

VueJS kann - wie Bootstrap - über CDN eingebunden werden (ich empfehle die Version 2.x):

```
<head>
...
<script src="https://cdn.jsdelivr.net/npm/vue@2"></script>
...
</head>
```

Für VueJS müssen wir unserem "Hauptbereich" ("Wurzel-div") eine ID vergeben. Außerdem fügt man danach (noch im body) ein <script>-Tag ein, in welches man den VueJS-Code schreibt:

```
<body>
  <div id="app" class="container">
    ...
  </div>
  <script>
    var app = new Vue({
      el: '#app',
      data: {
        schueler_name: '',
      }
    });
  </script>
</body>
```

Im Beispiel oben wird eine einfache Vue-Applikation erzeugt, die genau 1 Attribut (wie in Java) definiert: schueler\_name. Attribute werden immer im "data"-Bereich gemäß obigen Schemas definiert. Der Wert von "el" muss mit der ID des Haupt-divs übereinstimmen.

Attribute können nun direkt in HTML über geschwungene Klammern eingebettet werden, sodass dort immer der aktuelle Wert des Attributs eingefügt wird, z.B. so:

```
<h1>({ schueler_name })</h1>
```

Man kann auch schreibend auf Attribute zugreifen - hierfür muss man sie nur mit dem jeweiligen Eingabefeld über "v-model" verbinden:

```
<input v-model="schueler_name"/>
```

Nun wird ohne Verzögerung der aktuelle Wert der Textbox ins Attribut schueler\_name übertragen. Sie werden sozusagen automatisch synchronisiert - dies nennt sich auch "Data Binding".

Einer Vue-Applikation kann man auch Methoden hinzufügen, innerhalb welcher man (wie in Java) über "this" auf die Attribute zugreifen kann:

```
var app = new Vue({
  el: '#app',
  data: {
    schueler_name: '',
  },
  methods: {
    submit() {
      this.schueler_name = 'Test'
    }
  }
});
```

Methoden können über "v-on:click" über einen Button aufgerufen werden:

```
<button type="button" v-on:click="submit">Speichern</button>
```

Spezielle Methoden stellen "Computed Properties" dar - diese liefern einen Wert zurück (return) und können dann in HTML wie normale Attribute verwendet werden:

```
<h1>({ begruessung })</h1>
...
var app = new Vue({
  el: '#app',
  data: {
    schueler_name: '',
  },
  methods: {
    submit() {
      this.schueler_name = 'Test'
    }
  },
  computed: {
    begruessung() {
      return 'Hallo ' + this.schueler_name
    }
  }
});
```

Nähere Informationen und Beispiele findet ihr unter <https://v2.vuejs.org/v2/guide/>.

### Anforderungen

Erweitere das Schüler-Formular um einige dynamische Inhalte!

GKÜ:

- Der aktuell eingetragene Name des Schülers wird zusätzlich im h1 angezeigt ("Schülerdaten NAME")
- Der "Check"-Button wird zu einem "Speichern"-Button (ohne Eingabeüberprüfung)
  - Bei einem Klick auf den Speichern-Button werden alle Eingaben zurückgesetzt (ohne Eingabeüberprüfung)

GKV:

- Der "Check"-Button wird zu einem "Speichern"-Button (mit Eingabeüberprüfung)
  - Bei einem Klick wird eine VueJS-Methode aufgerufen, welche die aktuellen Eingaben überprüft
  - Bei Fehlern werden die Fehlermeldungen über VueJS rot in HTML angezeigt
  - Bei Erfolg wird eine Erfolgsmeldung angezeigt / ausgegeben und alle Eingaben zurückgesetzt (geleert)

EKÜ:

- Über eine Computed Property wird unter der Eingabe für die E-Mail-Adresse "live" eine Fehlermeldung angezeigt, wenn es sich um keine echte E-Mail-Adresse handelt

EKV:

- Über eine Computed Property wird das Häkchen "eigenberechtigt" automatisch an- und abgehakt - je nach eingegebenem Alter

### Beispiele

Fehlermeldungen Beispiel 1:

Schülerdaten Super Mario

Name:

Super Mario

Alter:

999

Eigenberechtigt:

☒

E-Mail:

super

Das Alter muss zwischen 0 und 100 liegen!

Falsche E-Mail-Adresse!

Speichern

Fehlermeldungen Beispiel 2:

Schülerdaten

Name:

Alter:

-100

Eigenberechtigt:

☐

E-Mail:

super@maria.at

Wahrheit ist das Alter zwischen 0 und 100 liegt!

Wird geladen...  
Das Alter muss zwischen 0 und 100 liegen!

Schülerdaten Super Mario

Name: Super Mario

Alter: 20

Eigenberechtigt: ☒

E-Mail: super@mario.at

Speichern

Erfolgsnachricht Beispiel:

Auf dieser Seite wird Folgendes angezeigt:  
Daten, die Daten wurden gespeichert!

OK

Name: Super Mario

Alter: 20

E-Mail: super@mario.at

Speichern

Zurückgesetzte Formularelemente nach Klick auf Speichern:

Schülerdaten

Name:


Alter:

Eigenberechtigt: ☐

E-Mail:

Speichern

### Abgabe

Ladet eure Abgabe hier mit dem Namen "klasse\_nachname.zip" (oder .html) hoch und kommt zu einem Abgabegespräch. Es ist **kein** Protokoll nötig.  
 [Beispielvue.html](#) 21. September 2022, 10:10 AM

Add submission

### Submission status

Submission status	No submissions have been made yet
Grading status	Graded
Last modified	-
Submission comments	<a href="#">▶ Comments (0)</a>

### Feedback

Grade	EK vollständig
Graded on	Wednesday, 11 January 2023, 9:02 AM
Graded by	DD Dolezal Dominik