

VUE.JS

Handout



Name: Elias Trewin

Matrikel-Nr: 3056844

Kurs: WWI16SCC

Dozent: Elias Henrich

<https://github.com/Joeeeyy/FlightVue>

Inhalt

Kurzbeschreibung.....	3
Einsatzzwecke.....	3
Einbindung von Vue	3
Erster-Start mit Vue.....	3
Template-System.....	3
Komponenten.....	4
Externe Komponenten	5
Funktionen in der entwickelten Applikation	6
Anzeige einer Weltkarte und der Flughäfen	6
Suche nach Flugverbindungen	6
Auswahl einer Flugroute	6
Ansicht der Flugroute auf der Karte.....	6
Dropdown-Auswahl der Flughafen	7
Auswahl der Flughafen über die Karte	7
Übersicht der ausgewählten Flüge als Buchung	7
Bedingungen.....	7
Funktionalität mit HTML und Javascript nativ nachbilden	8
Quellen	9

Kurzbeschreibung

Vue.js ist ein Javascript-Framework zur Erstellung von Single-Page-Webanwendungen auf dem Client. Vue ermöglicht dem Entwickler eine schnellere und modulare Entwicklung. Es werden verschiedene Web-Konzepte wie wiederverwendbare Komponenten genutzt, sodass ein Entwickler seinen Quellcode in mehreren Projekten verwenden kann.

Einsatzzwecke

Für kleine bis mittelgroße Anwendungen mit interaktiver Oberfläche lohnt es sich sehr schnell Frameworks wie Vue zu nutzen, da bspw. durch die Komponenten und das Template-System viel Arbeit eingespart werden kann.

Für große Anwendungen sollte man einen Blick auf Frameworks wie React oder Angular werfen.

Einbindung von Vue

Vue kann einfach über eine Zeile Code in das Projekt eingebunden werden. Es gibt zwei verschiedene Versionen. Einmal eine Version für die Entwicklung und einmal eine Version für die produktive Umgebung. In der Entwicklungsversion erhält der Entwickler hilfreiche Konsolen-Warnungen.

Entwickler-Version:

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

Produktive-Version:

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

Erster-Start mit Vue

Nach der Einbindung kann ein Element mit einer Id hinzugefügt werden, in welchem Vue initialisiert wird:

```
<div id="app">
  {{ message }}
</div>
```

Mit dem folgenden Javascript-Snippet wird Vue gestartet:

```
var app = new Vue({
  el: '#app',
  data: {
    message: 'Hello Vue!'
  }
})
```

Template-System

Mit Vue wird direkt ein Template-System ausgeliefert, wie man sie beispielsweise von anderen Projekten wie Angular oder MustacheJS kennt. Dadurch lässt sich die Logik sehr einfach und ansehnlich von der View trennen und eine bessere Lesbarkeit erreichen. Zudem können simple Bedingungen schon über die View erstellt werden und somit lässt sich aufwendiger Javascript-Code vermeiden.

HTML:

```
<div id="app-4">
  <ol>
    <li v-for="todo in todos">
      {{ todo.text }}
    </li>
  </ol>
</div>
```

Javascript:

```
var app4 = new Vue({
  el: '#app-4',
  data: {
    todos: [
      { text: 'Learn JavaScript' },
      { text: 'Learn Vue' },
      { text: 'Build something awesome' }
    ]
  }
})
```

Außerdem kann man sehr einfach Action-Listener erzeugen. Ein Klick-Listener ist sehr einfach zu erstellen:

HTML:

```
<div id="app">
  <a @click="meineFunktion()">Text</a>
</div>
```

Javascript:

```
var app = new Vue({
  el: '#app',
  methods: {
    meineFunktion() {
      alert("hallo");
    }
  }
});
```

Für weitere Informationen zum Template-System kann die Dokumentation von Vue genutzt werden:

<https://vuejs.org/v2/guide/>

Komponenten

In Vue können einfach wiederverwendbare Komponenten genutzt und erstellt werden. In dem erstellten Projekt wurde als Beispiel eine Button-Komponente erzeugt. Diese Komponente könnte jetzt in jedem anderen Projekt genutzt werden. Zudem kann sie ganz einfach über einen eigenen Tag „<my-button />“ überall im Projekt genutzt werden.

Eine Komponente an sich besitzt einen Tag-Namen, Properties, Objekte und Methoden. Über die Properties können Daten von einer Komponente an eine andere übergeben werden, sodass diese miteinander kommunizieren können.

HTML:

```
<my-button label="Buchen" :click="() => book()"></my-button>
```

Hier wird ein Button erzeugt mit dem Label „Buchen“ und der Funktion „Click“. Der Button soll also den Inhalt „Buchen“ haben und bei einem Klick die Funktion Book() ausführen (Information: „() => ...“ ist dasselbe wie „function() { ... }“ s. ECMA6). Vor dem „click“ steht noch „:“, da hier kein String übergeben werden soll, sondern eine Referenz auf den Inhalt des Attributes (Achtung: Wenn man nur book() eingeben würde, würde die Funktion sofort ausgeführt werden und man könnte die Funktion nicht weiter nutzen).

```
Vue.component('my-button', {
  props: ['label', 'click'],
  template: '<button @click="click()" type="button" class="action-button">{{label}}</button>',
});
```

Hier wird nun die Komponente „my-button“ erzeugt. Die Komponente hat die Properties „label“ und „click“ und das gegebene Template. In dem Template wird dann diese beiden Properties verwendet. Der Text des Buttons ist der Label und bei einem Klick auf den Button wird die übergebene Funktion ausgeführt.

Mit diesen Komponenten lassen sich Redundanzen vermeiden, spätere Änderungen werden direkt an allen Buttons durchgeführt und somit ergibt sich eine einfachere Wartbarkeit.

Externe Komponenten

Die Entwicklung kann auch rapide beschleunigt werden, indem man externe Komponenten einbindet. In der entwickelten Anwendung wird beispielsweise die Komponente „v-select“ verwendet. Dazu müssen nur die benötigte JS-Datei eingebunden und die Komponente registriert werden.

HTML:

```
<script src="https://unpkg.com/vue-select@latest"></script>
```

Javascript:

```
Vue.component('v-select', VueSelect.VueSelect);
```

Und schon kann die externe Komponente verwendet werden:

```
<v-select v-model="to" :options="airports"></v-select>
```

Funktionen in der entwickelten Applikation

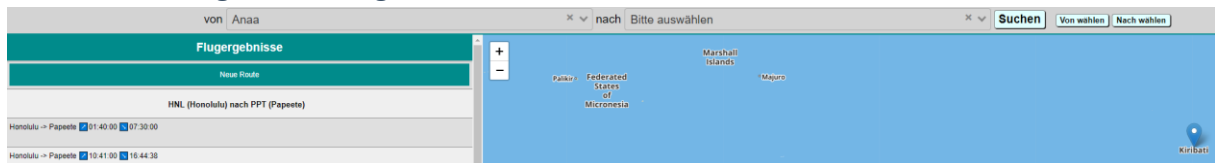
Die in der Vorlesung entwickelte Anwendung wurde im Rahmen der Arbeit auf Vue umgeschrieben. Dabei entstanden folgende Funktionalitäten:

Anzeige einer Weltkarte und der Flughäfen



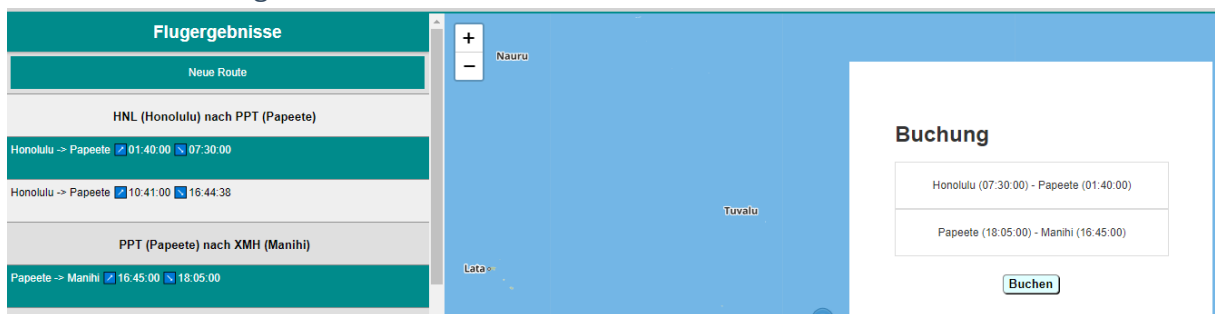
Nach dem Start der Applikation wird eine Weltkarte angezeigt und die Flughäfen der API werden per Marker angezeigt. Bei einem Hover über einen Marker wird der Name des Flughafens angezeigt.

Suche nach Flugverbindungen



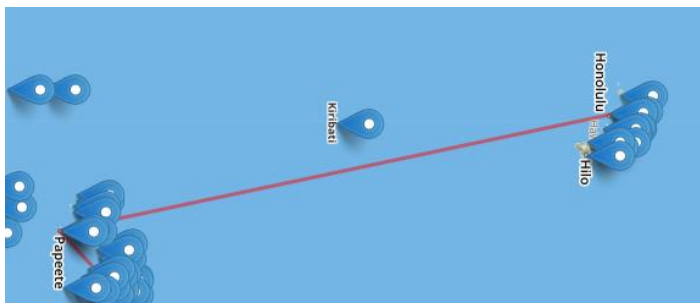
Die Applikation ermöglicht die Suche nach Flugrouten von einem Flughafen zu einem anderen.

Auswahl einer Flugroute



Durch einen Klick auf einen Flug ist es möglich eine Flugroute zu erstellen.

Ansicht der Flugroute auf der Karte



Nach Auswahl der Flugroute wird auf der Karte die Flugroute angezeigt.

Dropdown-Auswahl der Flughafen

The screenshot shows a form with two main sections: 'von' (from) and 'nach' (to). The 'von' section has a dropdown menu currently displaying 'Abaiang'. Below the dropdown, a list of other airports is visible: 'Abemama Atoll', 'Adak', 'Adelaide Airport', and 'Ahe Airport'. To the right of the dropdown are '+' and '-' buttons. The 'nach' section has a text input field with the placeholder 'Bitte auswählen' and two buttons: 'wählen' and 'Nach wählen'. The background of the interface features a map of the Pacific Ocean.

Die Flughäfen werden über die externe Komponente „v-select“ angezeigt und können ausgewählt werden. Durch die Auswahl werden die Ids der Flughäfen geholt und können an die Flugsuche übermittelt werden.

Auswahl der Flughafen über die Karte

This screenshot shows the same interface as before, but with a map overlay. A white rounded rectangle is centered on the map with the text 'Wähle den Abflugsflughafen x'. The map shows the Pacific Ocean. Above the map, there are buttons for 'Suchen', 'Von wählen', and 'Nach wählen'. On the left side, there is a sidebar with the heading 'Flugergebnisse' and a list of flight results, including 'Neue Route' and 'HNL (Honolulu) nach PPT (Papeete)'.

Nach dem Klick auf „Von wählen“ oder „Nach wählen“ erscheint ein Popup und man kann per Klick auf einen Marker den Flughafen auswählen.

Übersicht der ausgewählten Flüge als Buchung

This screenshot shows the interface with a booking summary on the right side. The summary is titled 'Buchung' and lists two flights: 'Honolulu (07:30:00) - Papeete (01:40:00)' and 'Papeete (18:05:00) - Manihi (16:45:00)'. Below the list is a 'Buchung' button. The left sidebar shows the flight results, including 'HNL (Honolulu) nach PPT (Papeete)' and 'PPT (Papeete) nach XMH (Manihi)'. The map in the background shows the Pacific Ocean with markers for Nauru, Tuvalu, and Lata.

Nach Auswahl einer Buchung erscheint auf der rechten Seite ein Buchungsfenster, welches die Flüge anzeigt und eine Buchung initiiert.

Bedingungen

Um die Flug-API zu nutzen muss ein CORS-Plugin installiert und aktiviert sein.

Funktionalität mit HTML und Javascript nativ nachbilden

Schon einfachste Anwendungsfälle sind mit nativem Javascript ohne Framework deutlich schlechter abzubilden. Mit Vue lässt sich das folgende Beispiel deutlich einfacher lesen, verstehen und modifizieren.

Natives Javascript:

```
<div id="content">

</div>
<script>
var persons = [
  {
    "name": "Peter",
    "age": 20
  },
  {
    "name": "Max",
    "age": 18
  }
];

for(var i = 0; i < persons.length; ++i) {
  document.getElementById("content").innerHTML +=
  ("<h1>"+persons[i].name+"</h1><p>"+persons[i].age+"</p>");
}
</script>
```

Mit Vue:

```
<div id="app">
  <div v-for="(person, index) in persons">
    <h1>{{person.name}}</h1>
    <p>{{person.age}}</p>
  </div>
</div>

<script>
var app = new Vue({
  el: '#app',
  data: {
    persons: [
      {
        "name": "Peter",
        "age": 20
      },
      {
        "name": "Max",
        "age": 18
      }
    ]
  }
})
</script>
```


Quellen

<https://de.wikipedia.org/wiki/Vue.js>

<https://vuejs.org/>

<https://vuejs.org/v2/guide/>

<https://github.com/sagalbot/vue-select>