



Experttalk

Vue - Grundlagen

Inhalt

- Initialisierung
- Komponente
- Data- & Input-binding
- List & v-for
- Methoden & Events
- v-if, v-else & data object



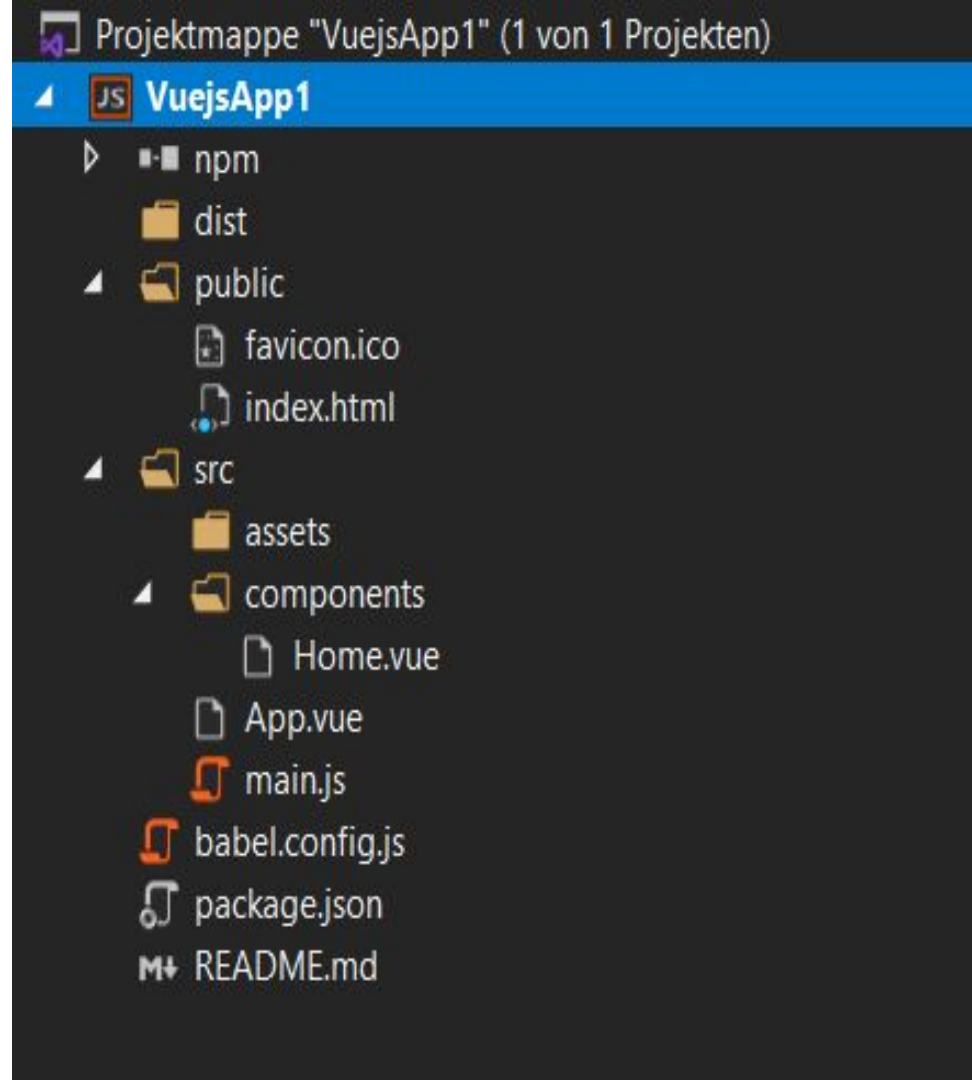
Live Coding

Initialisierung

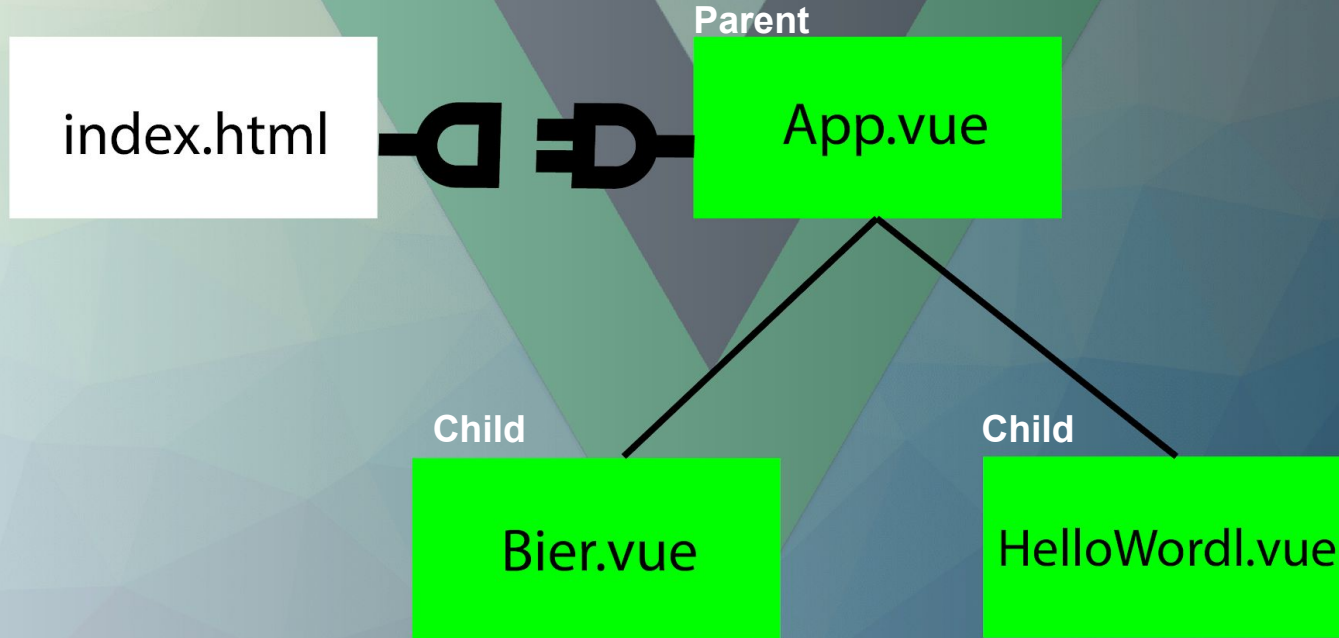


Initialisierung

- Readme.md
- PDF



GRUNDLAGEN - Struktur



Live Coding

Komponente



Bier-Komponente(child) einbinden

App.vue

```
<template>
  <div id="app">
    <HelloWorld msg="Welcome to Your Vue.js App"/>
    <Bier bier3 text="Reinheitsgebot" Parent Bier"/>
  </div>
</template>

<script>
import HelloWorld from './components/HelloWorld.vue'
import Bier from './components/Bier.vue'
export default {
  name: 'App',
  components: {
    HelloWorld,
    Bier,
  }
}
```

Bier.vue

```
<template>
  <div class="bier">
    <h1>Hello Child Bier</h1>
    <h2>{{sorte}}</h2>
  </div>
</template>

<script>
export default {
  name: 'Bier',
  props: ['bierText'],
  data() {
    return {
      sorte: "hefe",
    }
  },
}
```


Data und Input-binding: props VS. data

props:

- werden von parent -> child weitergereicht
- read-only und sollten nicht abgeändert werden

```
props: {  
  title: String,  
  likes: Number,  
  isPublished: Boolean,  
  commentIds: Array,  
  author: Object,  
  callback: Function,  
  contactsPromise: Promise // or any other constructor  
}
```

data:

- data = “Gedächtnis” jeder Komponente
- muss eine Funktion in der Komponente sein



Live Coding

The image features a blurred background of a computer monitor displaying code in a dark-themed editor. The code is color-coded with various colors like blue, green, and red. In the foreground, a pair of black-rimmed glasses is positioned horizontally, with the lenses reflecting the screen's content. The overall scene suggests a focus on programming and live coding.

Data-binding: props VS. data

App.vue

```
<template>
  <div id="app">
    <HelloWorld msg="Welcome to Your Vue.js App"/>
    <Bier bier-text="Reinheitsgebot" Parent Bier"/>
  </div>
</template>

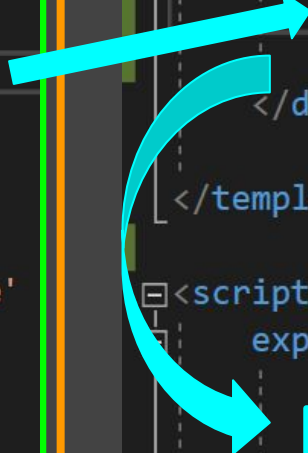
<script>
import HelloWorld from './components/HelloWorld.vue'
import Bier from './components/Bier.vue'

export default {
  name: 'App',
  components: {
    HelloWorld,
    Bier,
  }
}
```

Bier.vue


```
<template>
  <div class="bier">
    <h1>{{bierText}}</h1>
    <h2>{{sorte}}</h2>
  </div>
</template>

<script>
export default {
  name: 'Bier',
  props: ['bierText'],
  data() {
    return {
      sorte: "hefe",
    }
  },
}
```



Input-Binding

```
<template>  
  <div class="bier">  
    <h1>{{bierText}}</h1>  
    <input v-model="sorte" placeholder="Was darfst sein?" />  
    <h2>{{sorte}}</h2>  
  </div>  
</template>
```

A yellow rectangular box highlights the line `<input v-model="sorte" placeholder="Was darfst sein?" />`. A yellow curved arrow originates from the `sorte` attribute value and points to the `sorte` variable in the `<h2>{{sorte}}</h2>` line, illustrating the data binding.

Unordered list

List und v-for

```
<ul id="v-for-object" class="demo">
  <li v-for="value in object">
    {{ value }}
  </li>
</ul>
```

HTML

```
new Vue({
  el: '#v-for-object',
  data: {
    object: {
      title: 'How to do lists in Vue',
      author: 'Jane Doe',
      publishedAt: '2016-04-10'
    }
  }
})
```

JS

div

```
<div v-for="(value, name) in object">
  {{ name }}: {{ value }}
</div>
```

title: How to do lists in Vue
author: Jane Doe
publishedAt: 2016-04-10

Result:

- How to do lists in Vue
- Jane Doe
- 2016-04-10

Live Coding

List und v-for



List und v-for

Bier.vue

```
<template>
  <div class="bier">
    <h1>{{bierText}}</h1>
    <h2>{{sorte}}</h2>
    <div v-for="item in items" v-bind:key="item.id">
      {{item.marke}} - {{item.preis}}
    </div>
  </div>
</template>
```

```
<script>
  const imgBier = require("../assets/bier.jpg")
  const imgKO = require("../assets/ko.jpg")

  export default {
    name: 'Bier',
    props: ['bierText'],
    img: {
      imgBier,
      imgKO,
    },
    data() {
      return {
        sorte: "Helles",
        count: 0,
        items: [
          { marke: "Bitburger", preis: "9.2" },
          { marke: "Warsteiner", preis: "22.99" },
          { marke: "Licher", preis: "18.1" },
        ],
      }
    },
  }
</script>
```


Methoden und Events

- Methoden zur Komponente hinzufügen
- *computed* Property
 - Cache basiert

```
var vm = new Vue({  
  el: '#example',  
  data: {  
    message: 'Hello'  
  },  
  computed: {  
    // a computed getter  
    reversedMessage: function () {  
      // `this` points to the vm instance  
      return this.message.split('').reverse().join('')  
    }  
  }  
})
```

Event

```
<div id="example-1">  
  <button v-on:click="counter += 1">Add 1</button>  
  <p>The button above has been clicked {{ counter }} times.</p>  
</div>
```

```
var example1 = new Vue({  
  el: '#example-1',  
  data: {  
    counter: 0  
  }  
})
```

Live Coding

Methoden und Events



Methoden und Events

Bier.vue

```
<template>
  <div class="bier">
    <h1>{{bierText}}</h1>
    <input v-model="sorte" placeholder="Was darfst sein?" />
    <h2>{{sorte}}</h2>
    <div v-for=" item in items" v-bind:key="item.id">
      {{item.marke}} - {{item.preis}}
    </div>
    <button v-on:click="hochDenKolben()">Rein Zinken!</button>
  </div>
</template>
```

```
methods: {
  hochDenKolben() {
    this.count++
  }
},
```

```
</script>
```


Live Coding

v-if, v-else und
data object



v-if, v-else und data object

```
<template>
  <div class="bier">
    <h1>{{bierText}}</h1>
    <input v-model="sorte" placeholder="Was darfst sein?" />
    <h2>{{sorte}}</h2>
    <div v-for=" item in items" v-bind:key="item.id">
      {{item.marke}} - {{item.preis}}
    </div>
    
    
    <p id="warning" v-if="count === 7">Feierabend für Heute Kollege! Ich ruf dir ein Taxi!</p>
    <p v-else>{{count}}</p>
    <button v-on:click="hochDenKolben()">Rein den Zinken!</button>
  </div>
</template>
```

```
<script>
  const imgBier = require("../assets/bier.jpg")
  const imgKO = require("../assets/ko.jpg")

  export default {
    name: 'Bier',
    props: ['bierText'],
    data() {
      return {
        sorte: "Helles",
        count: 0,
        img: {
          imgBier,
          imgKO,
        },
        items: [
          { marke: "Bitburger", preis: "2.2" },
          { marke: "Warsteiner", preis: "22.99" },
          { marke: "Licher", preis: "18.1" },
        ],
      }
    },
  },
}
```

Vielen Dank für eure Aufmerksamkeit!



Feierabend für Heute, Kollege! Ich ruf' dir ein Taxi!