



Datenfluss, Kommunikation und Komponentenbau

Marc Teufel







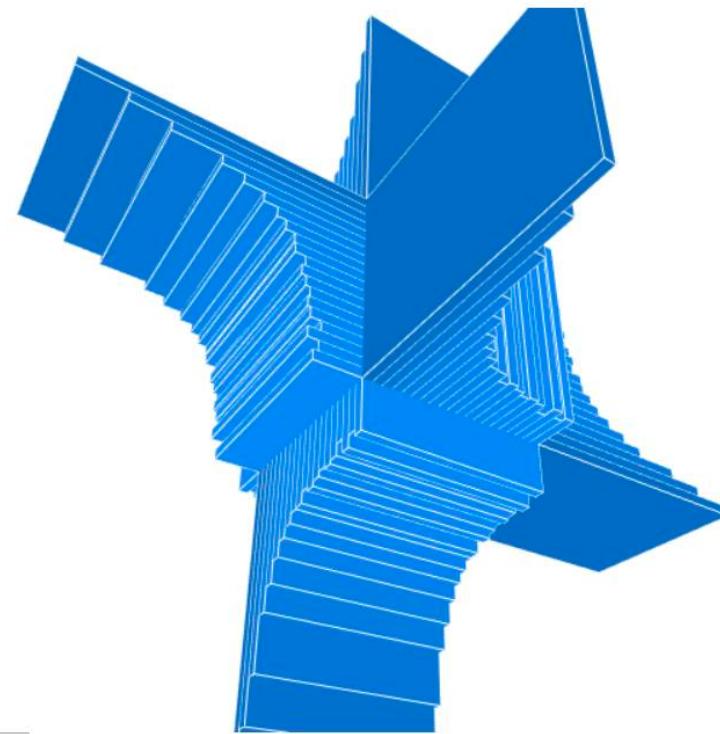
sidhub.org
0.0.3-SNAPSHOT



Gang I Den
Søren Lund (Jeff)
1993 Camelot



gang i den



connect disconnect

Gang I Den

Søren Lund (Jeff)

(c) 1993 Camelot

<http://www.sidhub.org>

hāma®



Teil 1



Komponentenbau



<> easy.html <

```
1  <!doctype html> <html lang="en">
2      <head>
3          <script src="https://unpkg.com/vue@3.0.0/dist/vue.runtime.global.js"></script>
4          <title>More Vue</title>
5      </head>
6      <body>
7          <div id="counter">
8              <h1>Counter: {{ counter }}</h1>
9          </div>
10     </body>
11     <script>
12         const Counter = {
13             data() {
14                 return { counter: 0 }
15             },
16             mounted() {
17                 setInterval(() => { this.counter++ }, 1000)
18             }
19         }
20         Vue.createApp(Counter).mount('#counter')
21     </script>
22 </html>
```



<https://unpkg.com/vue@next>

```
var Vue = (function (exports) {
  'use strict';

  /**
   * Make a map and return a function for checking if a key
   * is in that map.
   * IMPORTANT: all calls of this function must be prefixed with
   * \/*#\_\_PURE\_\_*\
   * So that rollup can tree-shake them if necessary.
   */
  function makeMap(str, expectsLowerCase) {
    const map = Object.create(null);
    const list = str.split(',');
    for (let i = 0; i < list.length; i++) {
      const value = list[i];
      if (expectsLowerCase) {
        map[value.toLowerCase()] = true;
      } else {
        map[value] = true;
      }
    }
    return map;
  }

  exports.makeMap = makeMap;
})((this.Vue || {}));
Object.defineProperty(exports, '__esModule', { value: true });
```

-
- vue.global.js
 - vue.runtime.global.js
 - vue.esm-browser.js
 - vue.runtime.esm-browser.js
 - vue.global.prod.js
 - vue.runtime.global.prod.js
 - vue.esm-browser.prod.js
 - vue.runtime.esm-browser.prod.js



<> easy.html <

```
1  <!doctype html> <html lang="en">
2      <head>
3          <script src="https://unpkg.com/vue@3.0.0/dist/vue.runtime.global.js"></script>
4          <title>More Vue</title>
5      </head>
6      <body>
7          <div id="counter">
8              <h1>Counter: {{ counter }}</h1> ← Template
9          </div>
10     </body>
11     <script>
12         const Counter = {
13             data() {
14                 return { counter: 0 } ← Komponente
15             },
16             mounted() {
17                 setInterval(() => { this.counter++ }, 1000)
18             }
19         }
20         Vue.createApp(Counter).mount('#counter')
21     </script>
22 </html>
```



index-1.html X

```
1  <!doctype html> <html lang="en">
2    <head>
3      <script src="https://unpkg.com/vue@3.0.0/dist/vue.runtime.global.js"></script>
4      <title>More Vue</title>
5    </head>
6    <body>
7      <div id="counter">
8        <h1>Counter: {{ counter }}</h1>
9      </div>
10     </body>
11     <script src="counter-1.js"></script> ←
12   </html>
13
```

JS counter-1.js X

```
1  const Counter = {
2    data() {
3      return { counter: 0 }
4    },
5    mounted() {
6      setInterval(() => { this.counter++ }, 1000)
7    }
8  }
9  Vue.createApp(Counter).mount('#counter')
10
```

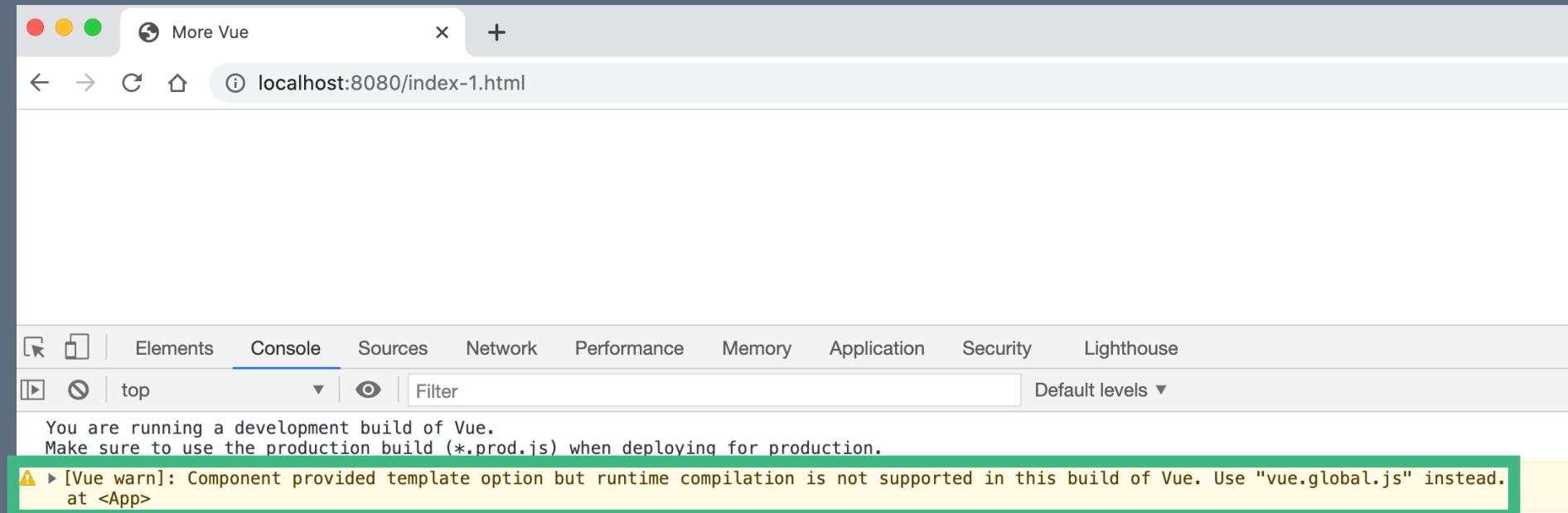


```
marcteufel@mac:~/web-prj/more-vue$ npx server --reload --browse  
npx: Installierte 1 in 0.943s
```

📁 Serving: /Users/marcteufel/web-prj/more-vue

🏡 Local: http://localhost:8080

🌐 Network: http://192.168.10.81:8080





index-1.html X

```
1  <!doctype html> <html lang="en">
2    <head>
3      <script src="https://unpkg.com/vue@3.0.0/dist/vue.runtime.global.js"></script>5
4      <title>More Vue</title>
5    </head>
6    <body>
7      <div id="counter">
8        <h1>Counter: {{ counter }}</h1>
9      </div>
10   </body>
11   <script src="counter-1.js"></script>
12 </html>
13
```

← Inline Templating!

JS counter-1.js X

```
1  const Counter = {
2    data() {
3      return { counter: 0 }
4    },
5    mounted() {
6      setInterval(() => { this.counter++ }, 1000)
7    }
8  }
9  Vue.createApp(Counter).mount('#counter')
10
```



index-1.html ×

```
1 <!doctype html> <html lang="en">
2   <head>
3     <script src="https://unpkg.com/vue@3.0.0/dist/vue.global.js"></script>
4     <title>More Vue</title>
5   </head>
6   <body>
7     <div id="counter">
8       <h1>Counter: {{ counter }}</h1>
9     </div>
10    </body>
11    <script src="counter-1.js"></script>
12  </html>
```

...

JS counter-1.js ×

```
1 const Counter = {
2
3   data() {
4     return { counter: 0 }
5   },
6   mounted() {
7     setInterval(() => { this.counter++ }, 1000)
8   }
9 }
10 Vue.createApp(Counter).mount('#counter')
11
12
13
14
```

index-2.html ×

```
1 <!doctype html> <html lang="en">
2   <head>
3     <script src="https://unpkg.com/vue@3.0.0/dist/vue.global.js"></script>
4     <title>More Vue</title>
5   </head>
6   <body>
7     <div id="counter"></div>
8   </body>
9   <script src="counter-2.js"></script>
10 </html>
```

JS counter-2.js ×

```
1 const Counter = {
2
3   template: '<h1>Counter: {{ counter }}</h1>',
4
5   data() {
6     return { counter: 0 }
7   },
8   mounted() {
9     setInterval(() => { this.counter++ }, 1000)
10   }
11 }
12 Vue.createApp(Counter).mount('#counter')
13
```

String
`Template
Literals`

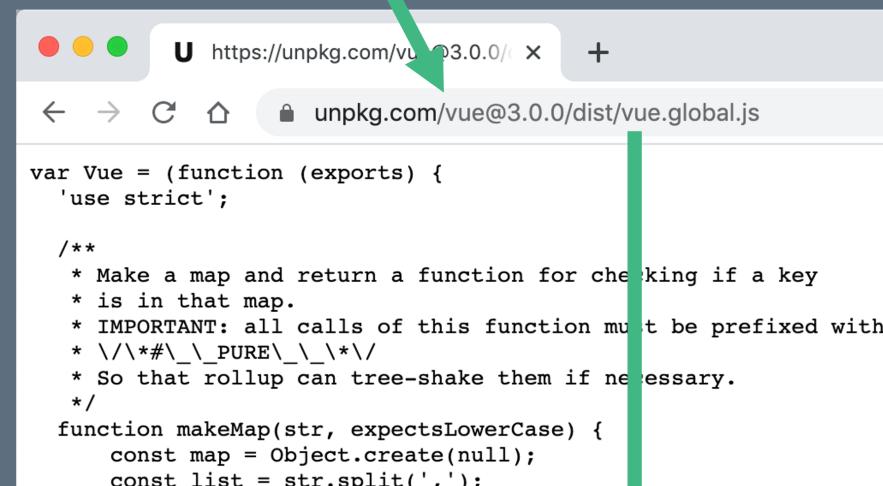


ESM

EcmaScript Modules

verwenden!

<https://unpkg.com/vue@next>



```
var Vue = (function (exports) {
  'use strict';

  /**
   * Make a map and return a function for checking if a key
   * is in that map.
   * IMPORTANT: all calls of this function must be prefixed with
   * `/\*\#\_\_PURE\_\_\*/`
   * So that rollup can tree-shake them if necessary.
   */
  function makeMap(str, expectsLowerCase) {
    const map = Object.create(null);
    const list = str.split(',');
    for (let i = 0; i < list.length; i++) {
      const value = list[i];
      if (expectsLowerCase) {
        exports[value] = true;
        map[value.toLowerCase()] = true;
      } else {
        exports[value] = true;
        map[value] = true;
      }
    }
    return map;
  }
  exports.makeMap = makeMap;
})((this.Vue || {}));
Object.defineProperty(exports, '__esModule', { value: true });
```

- vue.global.js
- vue.runtime.global.js
- vue.esm-browser.js
- vue.runtime.esm-browser.js
- vue.global.prod.js
- vue.runtime.global.prod.js
- vue.esm-browser.prod.js
- vue.runtime.esm-browser.prod.js



ohne

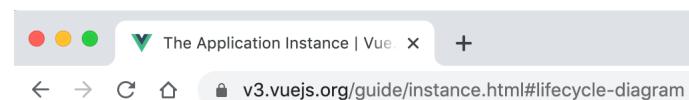
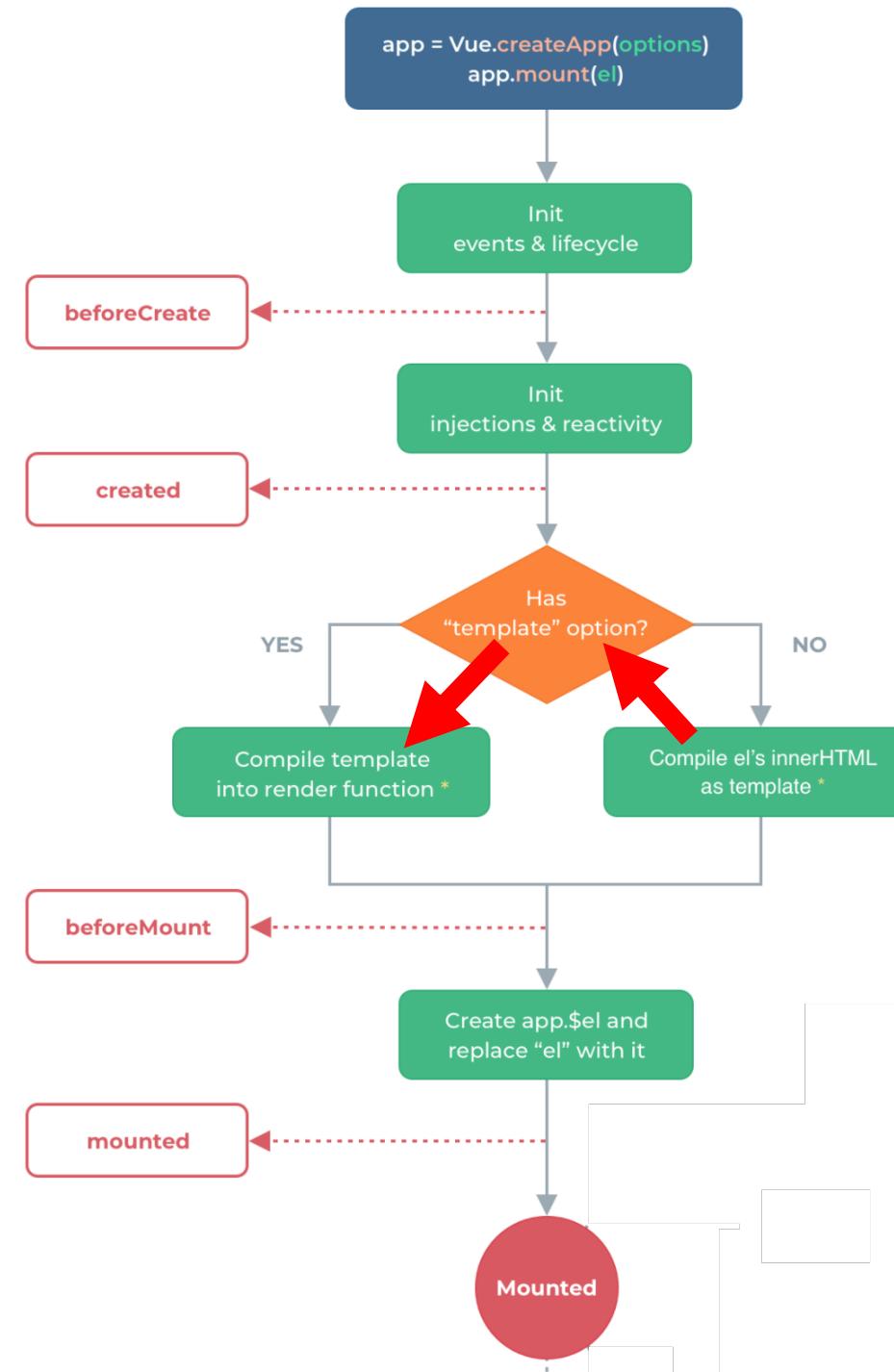
```
index-2.html × ...  
1  <!doctype html> <html lang="en">  
2    <head>  
3      <script src="https://unpkg.com/vue@3.0.0/dist/vue.global.js"></script>  
4      <title>More Vue</title>  
5    </head>  
6    <body>  
7      <div id="counter"></div>  
8    </body>  
9    <script src="counter-2.js"></script>  
10   </html>  
11  
12
```

```
JS counter-2.js × ...  
1  const Counter = {  
2  
3    template: '<h1>Counter: {{ counter }}</h1>',  
4  
5    data() {  
6      return { counter: 0 }  
7    },  
8    mounted() {  
9      setInterval(() => { this.counter++ }, 1000)  
10     }  
11   }  
12  Vue.createApp(Counter).mount('#counter')  
13
```

mit

```
index-esm.html × ...  
1  <!doctype html> <html lang="en">  
2    <head>  
3      <script src="counter-esm.js" type="module"></script>  
4      <title>More Vue</title>  
5    </head>  
6    <body>  
7      <div id="counter"></div>  
8    </body>  
9  </html>  
10
```

```
JS counter-esm.js × ...  
1  import { createApp }  
2  from 'https://unpkg.com/vue@3.0.0-rc.7/dist/vue.esm-browser.js'  
3  
4  const Counter = {  
5  
6    template: '<h1>Counter: {{ counter }}</h1>',  
7  
8    data() {  
9      return { counter: 0 }  
10     },  
11    mounted() {  
12      setInterval(() => { this.counter++ }, 1000)  
13     }  
14   }  
15  createApp(Counter).mount('#counter')  
16
```



Vue.js

Essentials

Installation

Introduction

The Application Instance

Creating an Instance

Data and Methods

Instance Lifecycle Hooks

Lifecycle Diagram

Template Syntax

Computed Properties and Watchers

Class and Style Bindings

Conditional Rendering

List Rendering

Event Handling

Form Input Bindings

Components Basics

Render-Funktionen?

Ganz zentrales Konzept von Vue!

Jede Vue-Komponente verfügt zur Laufzeit letztendlich über eine Render-Funktion

Entweder durch Kompilation oder selbstprogrammiert

Warum Render-Funktionen selber programmieren?

- komplette Kontrolle
- generische Komponenten, wo Logik bestimmt wie sich Template Markup zusammensetzt
- Komponenten die sehr viel Template Markup erfordern (z.B. lange Listen)
- ...



Eigene Render-Funktion schreiben

```
JS counter-esm2.js ×  
1 import { createApp, h }  
2   |   | from 'https://unpkg.com/vue@3.0.0-rc.7/dist/vue.runtime.esm-browser.js'  
3  
4 const Counter = {  
5  
6   render() {  
7     return h('h1', `Counter ${this.counter}`)  
8   },  
9  
10  data() {  
11    return { counter: 0 }  
12  },  
13  mounted() {  
14    setInterval(() => { this.counter++ }, 1000)  
15  }  
16}  
17 createApp(Counter).mount('#counter')
```

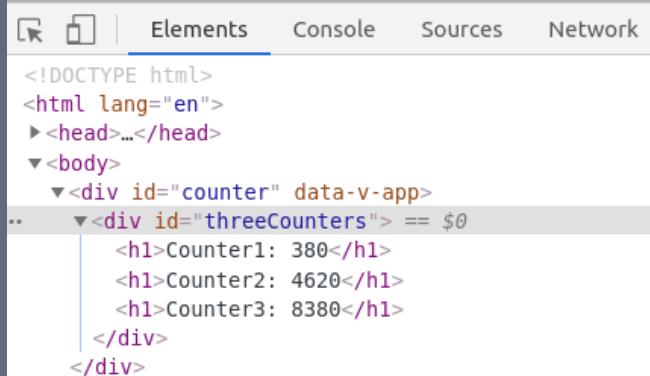


Komplexeres Markup

JS counter-esm3.js ×

```
1 import { createApp, h } from 'https://unpkg.com/vue@3.0.0-rc.7/dist/vue.runtime.esm-browser.js'
2
3 const Counter = {
4     render() {
5         return h('div', { id: 'threeCounters' },
6                 [ h('h1', `Counter1: ${this.counter}`),
7                   h('h1', `Counter2: ${this.counter2}`),
8                   h('h1', `Counter3: ${this.counter3}`) ] )
9     },
10    data() {
11        return { counter: 0,
12                counter2: 5000,
13                counter3: 8000
14            }
15    },
16    mounted() {
17        setInterval(() => {
18            this.counter++
19            this.counter2--
20            this.counter3++
21        }, 1000)
22    }
23 }
24 createApp(Counter).mount('#counter')
```

Counter1: 380
Counter2: 4620
Counter3: 8380



The screenshot shows the browser's developer tools with the "Elements" tab selected. The DOM tree is displayed, starting with the DOCTYPE declaration and the HTML root element. Inside the body, there is a div with the id "counter" and the data-v-app attribute. This div contains another div with the id "threeCounters". The "threeCounters" div contains three h1 elements, each displaying a counter value: "Counter1: 380", "Counter2: 4620", and "Counter3: 8380".

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body>
    <div id="counter" data-v-app>
      <div id="threeCounters"> == $0
        <h1>Counter1: 380</h1>
        <h1>Counter2: 4620</h1>
        <h1>Counter3: 8380</h1>
      </div>
    </div>
  </body>
</html>
```



Wiederverwendbare Komponenten bauen

component.html ×

```
1  <!doctype html>
2  <html lang="en">
3      <head>
4          <script src="component.js" type="module"></script>
5          <title>More Vue</title>
6      </head>
7      <body>
8          <div id="app">
9              <counter :no="1" :count-forward="false" :initial-value="500" />
10             <counter :no="2" :count-forward="false" :initial-value="200"></counter>
11             <counter :no="3" :count-forward="true" :initial-value="0"></counter>
12         </div>
13     </body>
14 </html>
```

Tipp:
So lieber
nicht!





Wiederverwendbare Komponenten bauen

```
<!doctype html>
<html lang="en">
  <head>
    <script src="component.js" type="module"></script>
    <title>More Vue</title>
  </head>
  <body>
    <div id="app">
      <counter :no="1" :count-forward="false" :initial-value="500"></counter>
      <counter :no="2" :count-forward="false" :initial-value="200"></counter>
      <counter :no="3" :count-forward="true" :initial-value="0"></counter>
    </div>
  </body>
</html>
```

JS component.js ×

```
import { createApp, h } from 'https://unpkg.com/vue@3.0.0/dist/vue.esm-browser.js'
const app = createApp({})

app.component('counter', {
  props: {
    no: Number,
    countForward: Boolean,
    initialValue: Number
  },
  render() {
    return h('h1', `Counter ${this.no}: ${this.counter}`)
  },
  data() {
    return { counter: this.initialValue }
  },
  mounted() {
    setInterval(() => {
      if (this.countForward) {
        this.counter++
      } else {
        this.counter--
      }
    }, 1000)
  }
})
app.mount('#app')
```





Single File Components

JSX

Single File Components/SFC (.vue) und JSX gibt's nur kompiliert!

- Werden zur Entwicklungs- beziehungsweise Buildzeit optimiert und kompiliert
- ...in eine Render-Funktion!
- Daher spätestens ab jetzt ein Projekt-Setup erforderlich:
 - from scratch: `npm init -y`, package.json selbst aufbauen, Wepack ...
 - Vue CLI

*„As a JavaScript developer in 2019 I can relate.
We have this new JavaScript module system (ESM) that runs
Natively on the web. Yet we continue to use bundlers for every
single thing we build. Why?”*

Fred K. Schott

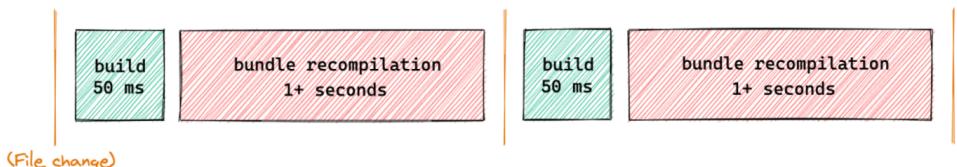
ab Vue 3, Entwicklung ohne Bundler auf Basis von ESM:

- Snowpack <https://www.snowpack.dev/> (von Fred K. Schott)
- Vite <https://github.com/vitejs/vite> (von Evan You)

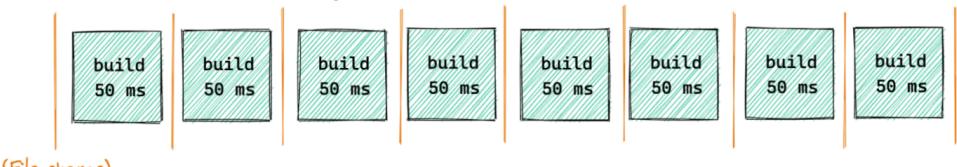
Warum Vite oder Snowpack verwenden?

- Zur Entwicklungszeit kein Bundling = schnellere Entwicklung gerade in größeren Projekten
- Bei Änderung einer Datei (egal ab .vue oder .js), wird nur diese vom mitgelieferten Dev-Server kompiliert und bereitgestellt
- Auslieferung an Browser im ESM-Format, native ESM-Imports = Debugging des Codes direkt im Browser viel einfacher möglich
- Productionbuild mit Vite: **Rollup**, eigene Anpassung und Konfiguration möglich
- Productionbuild mit Snowpack: Über Plugins - Bundler wie **Parcel** oder **Webpack** können eingebunden werden

Bundled (ex: Webpack)



Unbundled (Snowpack)



Unbundled development has several advantages over the traditional bundled development approach:

- Single-file builds are fast.
- Single-file builds are deterministic.
- Single-file builds are easier to debug.
- Project size doesn't affect dev speed.
- Individual files cache better.

That last point is key: **Every file is built individually and cached indefinitely**. Your dev environment will never build a file more than once and your browser will never download a file twice (until it changes). This is the real power of unbundled development.

Teil 2



Datenfluss, Kommunikation



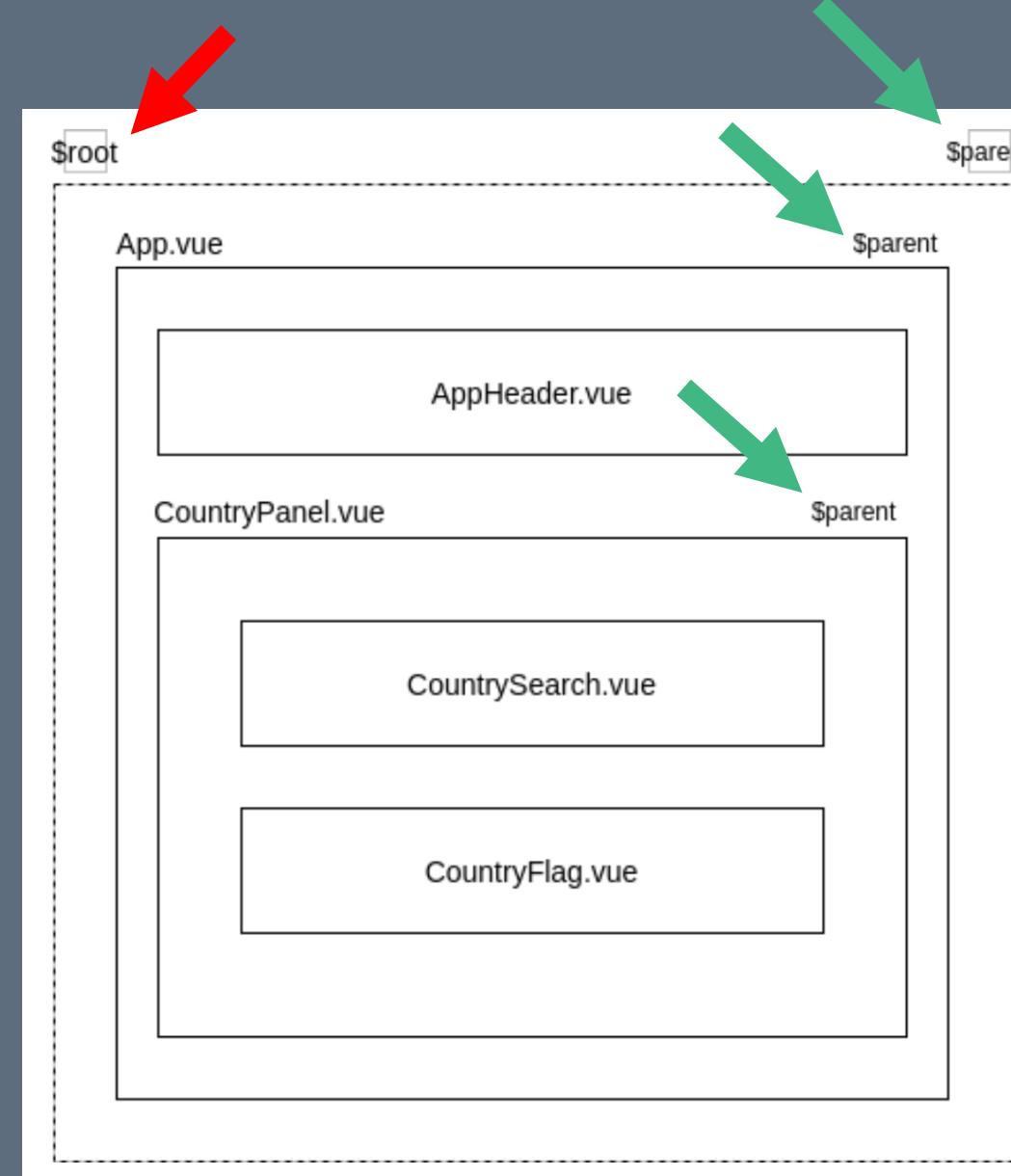
Aller guten Dinge sind DREI!



- \$root/\$parent
- Durchreichen → Props/Emit
- Rumreichen → Globale Stores, State Management
- Reinreichen → Provide/Inject

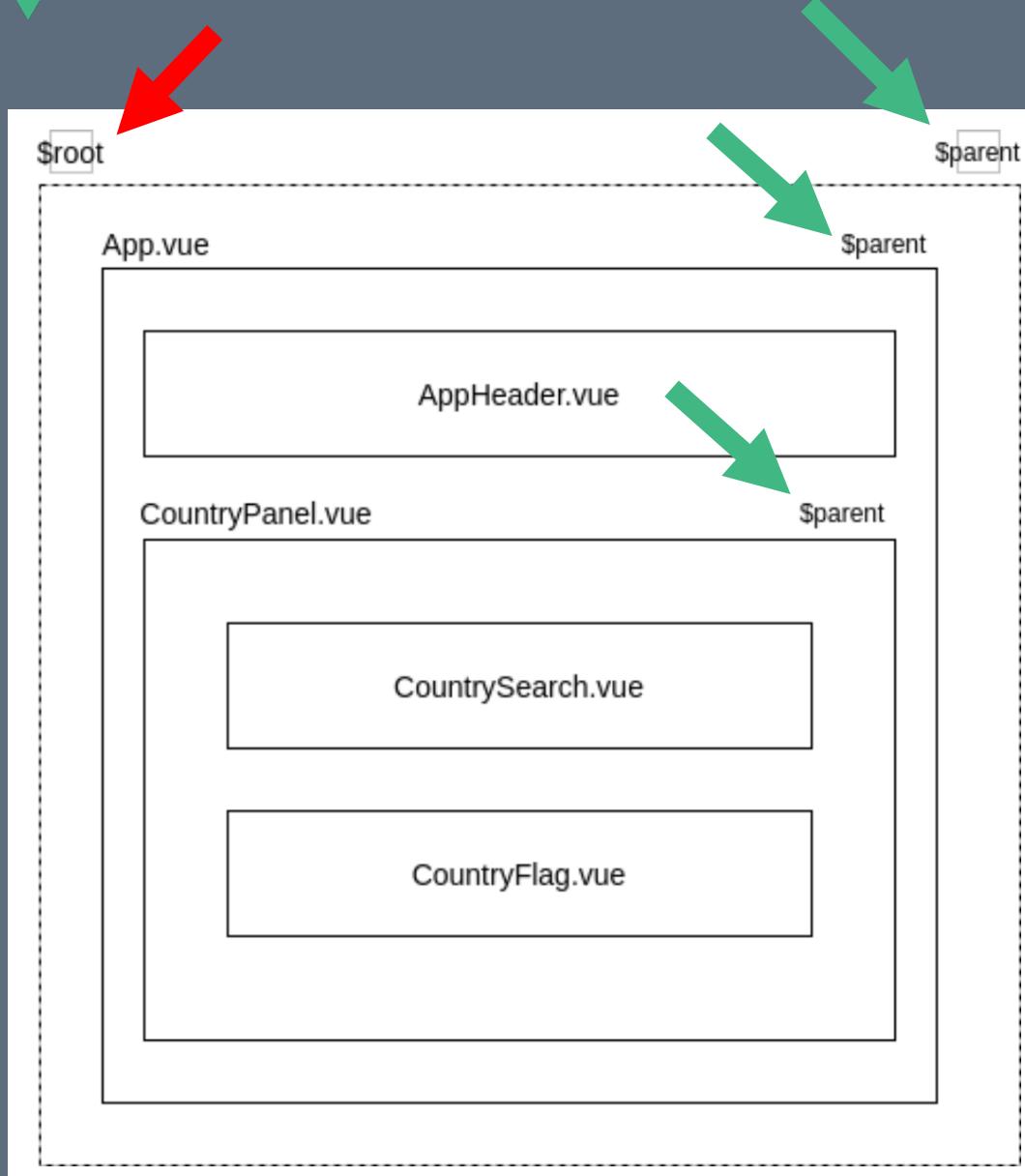


\$root/\$parent





\$root/\$parent



```
JS main.js ×  
1 import { createApp, h } from 'vue'  
2 import Antd from 'ant-design-vue'  
3 import App from './App.vue'  
4 import 'ant-design-vue/dist/antd.css'  
5  
6  
7 const rootInstance = createApp({  
8   data() {  
9     return { initialCountry: 'ro' }  
10   },  
11   render() {  
12     return h(App)  
13   }  
14 }  
15 )  
16 rootInstance.config.productionTip = false;  
17 rootInstance.use(Antd)  
18 rootInstance.mount('#app')
```

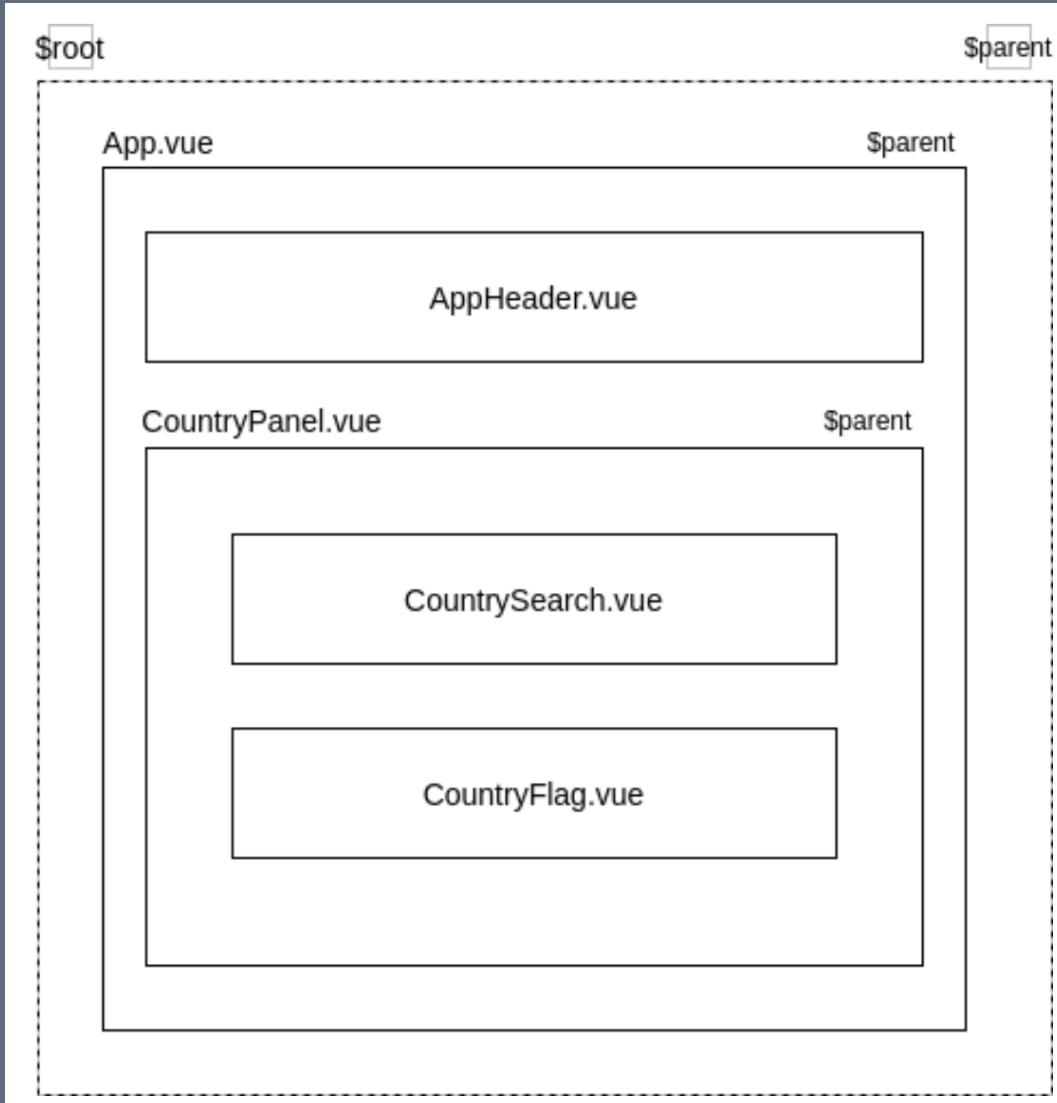
```
▼ AppHeader.vue ×  
1 <template>  
2   <div>  
3     <h1>{{ title }} --> Initial Country was {{ this.$root.initialCountry }}</h1>  
4   </div>  
5 </template>  
6  
7 <script>  
8 export default {  
9   name: "AppHeader",  
10  props: {  
11    title: String  
12  }  
13 }  
14 </script>
```

Welcome to Flag Search --> Initial Country was ro

Search for a country here



\$root/\$parent



```
<template>
  <div id="app">
    <AppHeader title="Welcome to Flag Search" />
    <CountryPanel>
      <CountrySearch />
      <CountryFlag :country="country" />
    </CountryPanel>
  </div>
</template>

<script>
import AppHeader from "./components/AppHeader";
import CountrySearch from "./components/CountrySearch";
import CountryPanel from "./components/CountryPanel";
import CountryFlag from "./components/CountryFlag";

export default {
  name: "App",
  data() {
    return {
      country: this.$root.initialCountry
    }
  },
  components: [
    AppHeader,
    CountryPanel,
    CountrySearch,
    CountryFlag
  ],
  methods: {
    onSelection(selectedValue) {
      this.$parent.$parent.country = selectedValue;
    }
  }
}
</script>
```

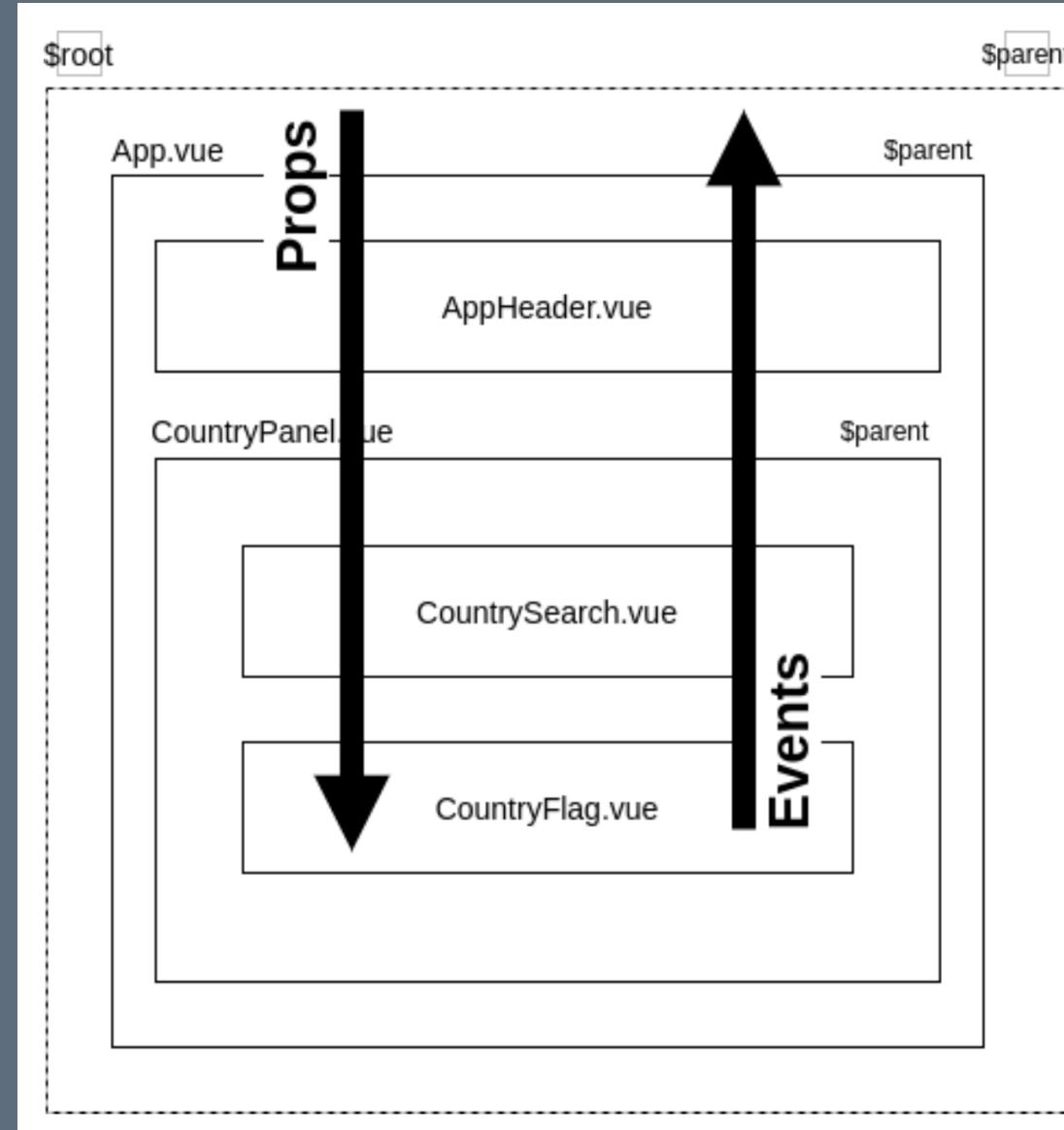
Welcome to Flag Search --> Initial Country was ro

Germany (Berlin)



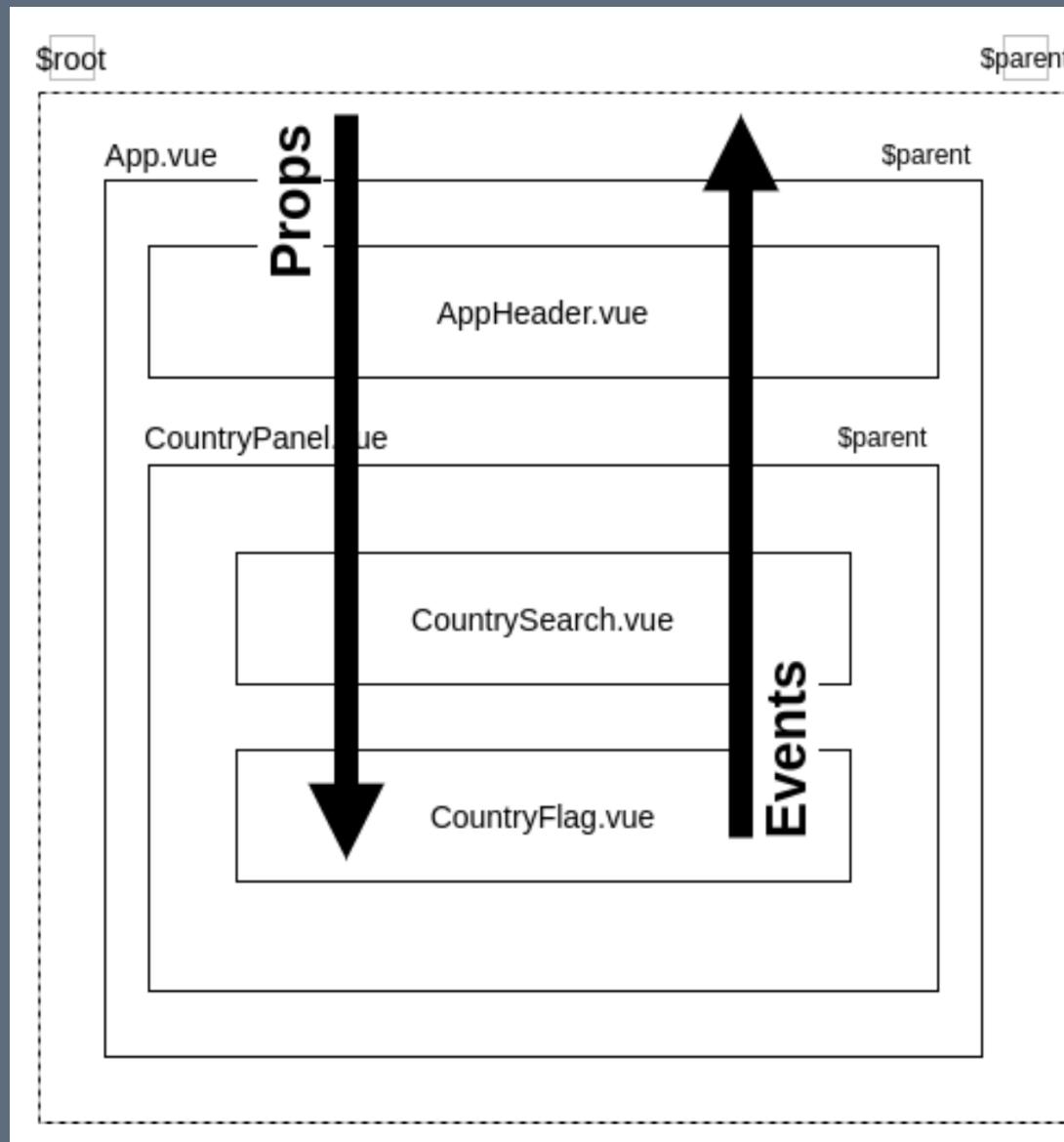


Daten durchreichen mit dem Standard-Weg: Props/Emit





Daten durchreichen mit dem Standard-Weg: Props/Emit

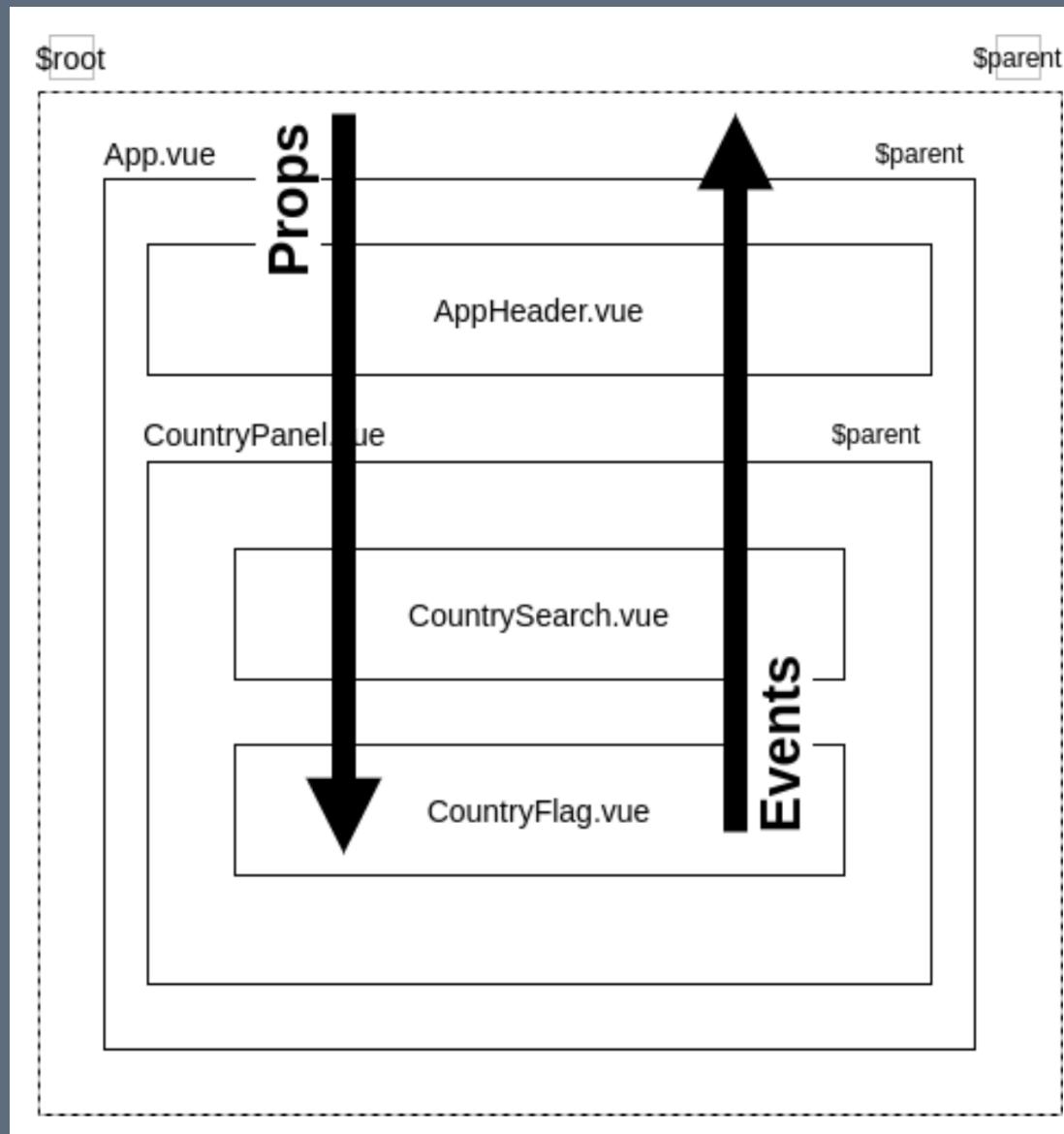


```
You, a few seconds ago | 1 author (You)
1 <template>
2   <div id="app">
3     <AppHeader title="Welcome to Flag Search" />
4     <CountryPanel>
5       <CountrySearch @country-changed="countryChanged" />
6       <CountryFlag :country="country" />
7     </CountryPanel>
8   </div>
9 </template>
10
11 <script>
12   import AppHeader from "./components/AppHeader.vue";
13   import CountrySearch from "./components/CountrySearch.vue"
14   import CountryPanel from "./components/CountryPanel.vue"
15   import CountryFlag from "./components/CountryFlag.vue";
16
17
18   export default {
19     name: "App",
20     data() {
21       return {
22         country: 'gb'
23       }
24     },
25     methods: {
26       countryChanged(selectedCountry) {
27         console.log('event angekommen: ' + selectedCountry)
28         this.country = selectedCountry
29       }
30     },
31     components: {
32       AppHeader,
33       CountryPanel,
34       CountrySearch,
35       CountryFlag
36     }
37   };
38 </script>
```

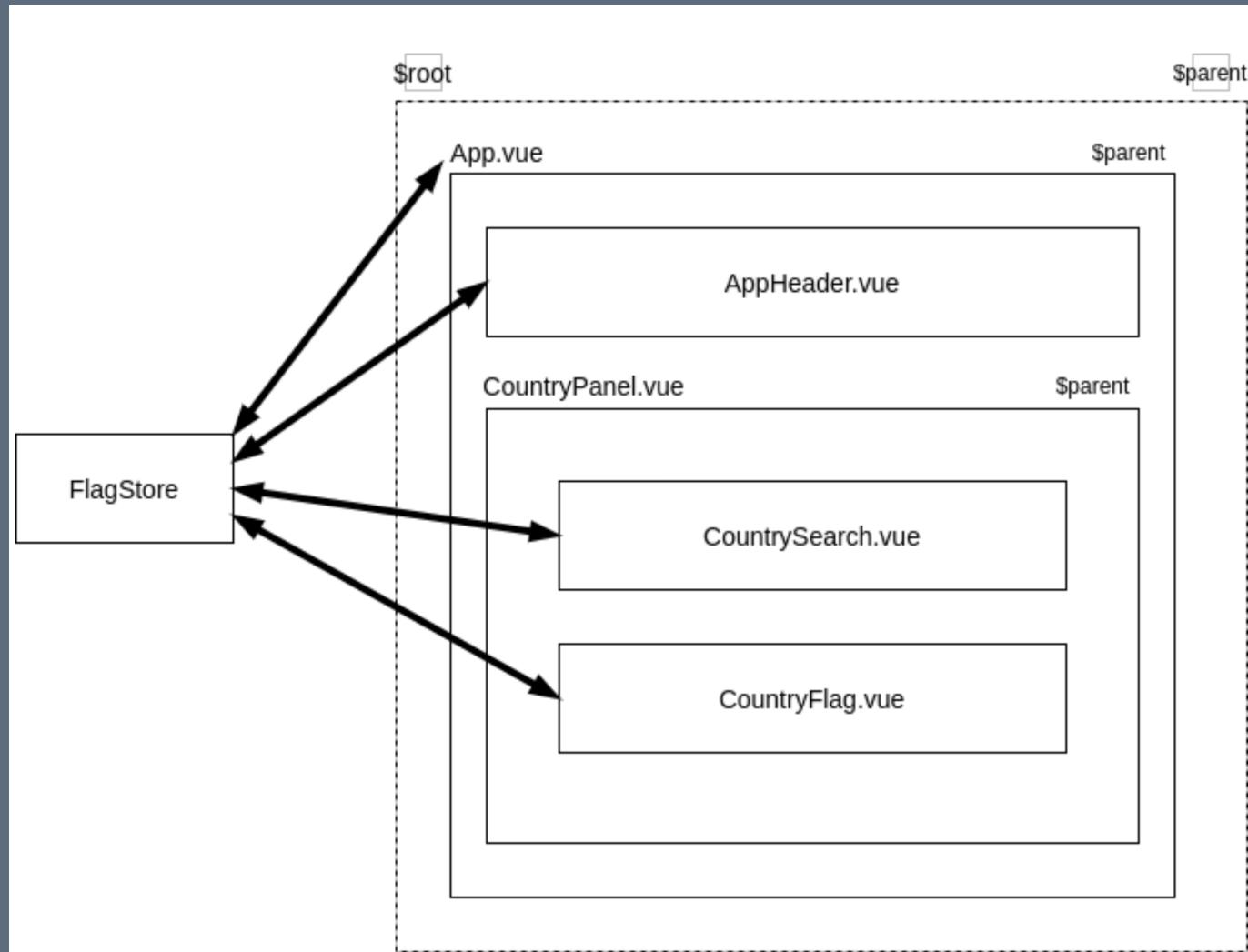
The code shows the implementation of the **Props** and **Events** pattern:

- Props:** The **CountryFlag** component is defined with a prop named `:country="country"`.
- Events:** The **CountrySearch** component emits an event named `@country-changed="countryChanged"` when a country is selected.
- Event Handler:** In the **App.vue** script, a method `countryChanged(selectedCountry)` is defined to handle the received event. It logs the selected country to the console and updates the **country** state variable.

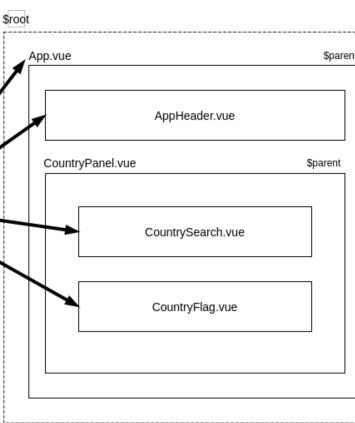
Daten durchreichen mit dem Standard-Weg: Props/Emit



Daten rumreichen mit Stores



Daten rumreichen mit Stores



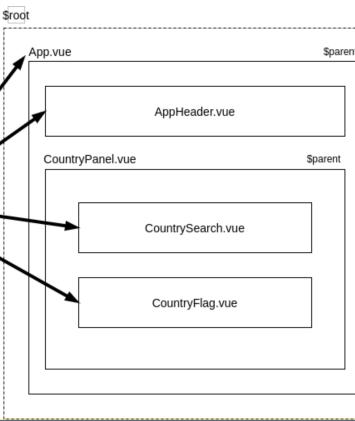
JS FlagStore.js ×

```
1  const createFlagStore = () => {
2
3    const url = 'https://restcountries.eu/rest/v2'
4
5    const queryByName = async (search) => {
6      let response = await fetch(`/${url}/name/` + search)
7      return await response.json()
8    }
9
10   let options = []
11   let selectedCountry = 'it'
12
13   return {
14
15     getOptions: () => options,
16     getSelectedCountry: () => selectedCountry,
17
18     doSearch: async(search) => {
19       if (search === "") {
20         return
21       }
22       let json = await queryByName(search)
23       options = await json.map(e => {
24         return { text: `${e.name} (${e.capital})`,
25                 value: e.alpha2Code.toLowerCase() }
26       })
27     }
28   }
29 }
30
31 const flagStore = createFlagStore()
32
33 export default flagStore
```

CountrySearch.vue ×

```
6  <div>
7    <a-select>
8      :filter-option="false"
9      :default-active-first-option="false"
10     :show-arrow="false"
11     @search="onSearch"
12     @change="onSelection">
13       <a-select-option v-for="o in options" :key="o.value">
14         {{ o.text }}
15       </a-select-option>
16     </a-select>
17   </div>
18 </template>
19
20 <script>
21
22 import { ref } from 'vue'
23 import flagStore from '../stores/FlagStore'
24
25 export default {
26   name: "CountrySearch",
27   setup() {
28     const options = ref([])
29     return { options }
30   },
31   methods: {
32     onSearch(search) {
33       flagStore.doSearch(search)
34       this.options = flagStore.getOptions();
35     },
36     onSelection(selectedValue) {
37       // TODO
38     }
39   }
40 </script>
41
```

Observer Pattern hilft!



JS FlagStore.js ×

```
1 const createFlagStore = () => {
2
3     const url = 'https://restcountries.eu/rest/v2'
4
5     const queryByName = async (search) => {
6         let response = await fetch(` ${url}/name/` + search)
7         return await response.json()
8     }
9
10    let options = []
11    let selectedCountry = 'it'
12    let subscribers = []
13
14    return {
15
16        getOptions: () => options,
17        getSelectedCountry: () => selectedCountry,
18
19        doSearch: async(search) => {
20            if (search === "") {
21                return
22            }
23            let json = await queryByName(search)
24            options = await json.map(e => {
25                return { text: `${e.name} (${e.capital})`,
26                        value: e.alpha2Code.toLowerCase() }
27            })
28        },
29
30        subscribe: (fn) => {
31            subscribers.push(fn)
32        },
33        countrySelected: (sel) => {
34            selectedCountry = sel;
35            subscribers.forEach( s =>s())
36        }
37
38    }
39
40}
41 const flagStore = createFlagStore()
42 export default flagStore
```

▼ App.vue ×

```
1 <template>
2     <div id="app">
3         <AppHeader title="Welcome to Flag Search" />
4         <CountryPanel>
5             <CountrySearch />
6             <CountryFlag :country="country" />
7         </CountryPanel>
8     </div>
9     </template>
10
11 <script>
12
13 import AppHeader from "./components/AppHeader.vue"
14 import CountrySearch from "./components/CountrySearch.vue"
15 import CountryPanel from "./components/CountryPanel.vue"
16 import CountryFlag from "./components/CountryFlag.vue"
17 import flagStore from "./stores/FlagStore"
18 import { ref } from 'vue'
19
20 export default {
21     name: "App",
22     setup() {
23
24         const country = ref(flagStore.getSelectedCountry())
25
26         flagStore.subscribe( () => country.value = flagStore.getSelectedCountry() )
27
28         return { country }
29     },
30     components: {
31         AppHeader,
32         CountryPanel,
33         CountrySearch,
34         CountryFlag
35     }
36 };
37 </script>
```

```
graph LR; A[FlagStore.js] --> B[App.vue]; A --> C[App.vue];
```



Dependency Injection mit Provide/Inject

Normalerweise reichen wir Daten mit Props durch die Komponentenhierarchie

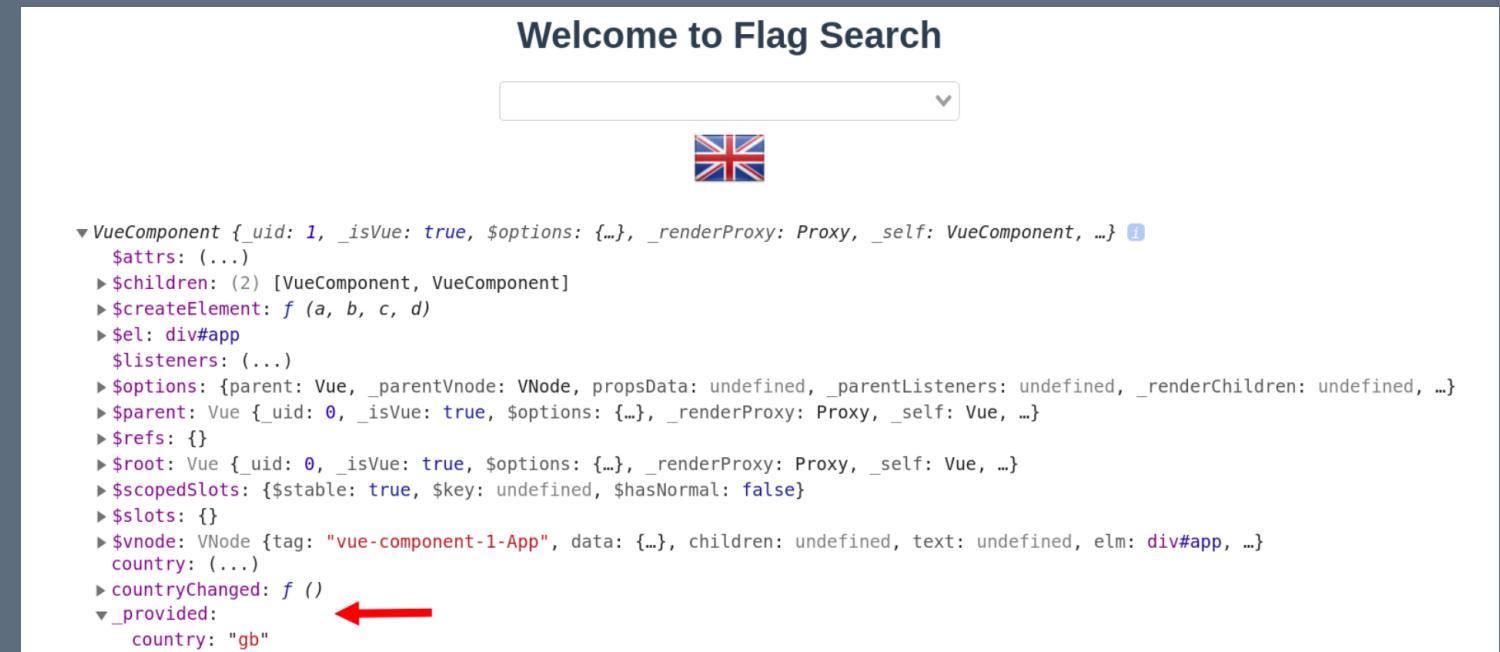
Was ist wenn die Hierarchie sehr tief ist? Wirklich überall durchreichen?

Hier kann Provide/Inject helfen!

→ Daten von einem beliebigen Punkt in der Komponentenhierarchie nach oben oder unten weiterreichen.

Dependency Injection mit Provide/Inject

```
App.vue X
1  <template>
2    <div id="app">
3      <AppHeader title="Welcome to Flag Search" />
4      <CountryPanel>
5        <CountrySearch />
6        <CountryFlag />
7      </CountryPanel>
8    </div>
9  </template>
10
11 <script>
12
13 import AppHeader from "./components/AppHeader.vue"
14 import CountrySearch from "./components/CountrySearch.vue"
15 import CountryPanel from "./components/CountryPanel.vue"
16 import CountryFlag from "./components/CountryFlag.vue"
17 import { ref, provide } from 'vue'
18
19 export default {
20   name: "App",
21   setup() {
22     const country = ref('us')
23     provide('country', country)
24     return { country }
25   },
26   components: {
27     AppHeader,
28     CountryPanel,
29     CountrySearch,
30     CountryFlag
31   }
32 }
33 </script>
```



Dependency Injection mit Provide/Inject

The image shows two code editors side-by-side, illustrating the use of dependency injection in Vue.js via the `provide` and `inject` APIs.

App.vue:

```
1 <template>
2   <div id="app">
3     <AppHeader title="Welcome to Flags" />
4     <CountryPanel>
5       <CountrySearch />
6       <CountryFlag />
7     </CountryPanel>
8   </div>
9 </template>
10 <script>
11 import AppHeader from "./components/AppHeader.vue"
12 import CountrySearch from "./components/CountrySearch.vue"
13 import CountryPanel from "./components/CountryPanel.vue"
14 import CountryFlag from "./components/CountryFlag.vue"
15 import { ref, provide } from 'vue'
16
17 export default {
18   name: "App",
19   setup() {
20     const country = ref('us')
21     provide('country', country)
22     return { country }
23   },
24   components: {
25     AppHeader,
26     CountryPanel,
27     CountrySearch,
28     CountryFlag
29   }
30 };
31 </script>
```

CountrySearch.vue:

```
1 <template>
2   <a-select
3     show-search
4     placeholder="Search for a country here"
5     style="width: 500px"
6     :filter-option="false"
7     :default-active-first-option="false"
8     :show-arrow="false"
9     @search="onSearch"
10    @change="onSelection">
11      <a-select-option v-for="o in options" :key="o.value">
12        {{ o.text }}
13      </a-select-option>
14    </a-select>
15  </div>
16 </template>
17 <script>
18 import { useSearch } from '../use/useCountries'
19 import { inject } from 'vue'
20
21 export default {
22   name: "CountrySearch",
23   setup() {
24     const country = inject('country')
25     return { country,
26       ...useSearch()
27     },
28     methods: {
29       onSelection(selectedValue) {
30         this.country = selectedValue
31       }
32     }
33   }
34 </script>
```

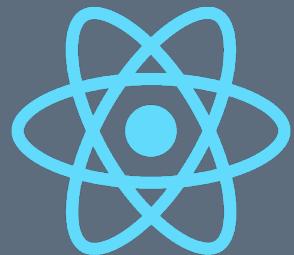
Three green arrows point from the `provide` call in `App.vue` to the `country` dependency in the `CountrySearch` component's `setup` function. This illustrates how `provide` makes a value available to child components via `inject`.

Dependency Injection mit Provide/Inject

```
App.vue X
1 <template>
2   <div id="app">
3     <AppHeader title="Welcome to Flag Search" />
4     <CountryPanel>
5       <CountrySearch />
6       <CountryFlag />
7     </CountryPanel>
8   </div>
9 </template>
10 <script>
11
12 import AppHeader from "./components/AppHeader.vue"
13 import CountrySearch from "./components/CountrySearch.vue"
14 import CountryPanel from "./components/CountryPanel.vue"
15 import CountryFlag from "./components/CountryFlag.vue"
16 import { ref, provide } from 'vue'
17
18 export default {
19   name: "App",
20   setup() {
21
22     const country = ref('us')
23     provide('country', country)
24     return { country }
25   },
26   components: {
27     AppHeader,
28     CountryPanel,
29     CountrySearch,
30     CountryFlag
31   }
32 };
33 </script>
```

```
CountrySearch.vue X
4   <input type="text" placeholder="Search for a country here" style="width: 500px" :filter-option="false" :default-active-first-option="false" :show-arrow="false" @search="onSearch" @change="onSelection">
5     <a-select-option v-for="o in options" :key="o.value" :label="o.text">
6       </a-select-option>
7     </a-select>
8   </div>
9 </template>
10 <script>
11
12 import { useSearch } from '../composables/useSearch'
13 import { inject } from 'vue'
14
15 export default {
16   name: "CountrySearch",
17   setup() {
18     const country = inject('country')
19     return { country, ...useSearch() }
20   },
21   methods: {
22     onSelection(selectedValue) {
23       this.country = selectedValue
24     }
25   }
26 }
27 </script>
```

```
CountryFlag.vue X
1 <template>
2   <div>
3     
4   </div>
5 </template>
6
7 <script>
8 import { computed, inject } from 'vue'
9
10 export default {
11   name: "Flag",
12   props: {
13     country: String
14   },
15   setup(props) {
16
17     //const flagCss = computed( () => "flag flag--" + props.country)
18
19     const flagCss = computed( () => "flag flag--" + inject('country').value)
20
21     return { flagCss }
22   }
23 }
24 </script>
```



Sie suchen einen auf Sie zugeschnittenen Workshop
oder Hilfe und Unterstützung in Ihren
Vue/React/JavaScript-Projekten ?

Ich unterstütze Sie gern!

Kontaktieren Sie mich:



Mail:

teufel.marc@gmail.com



Twitter: [@marcteufel](https://twitter.com/marcteufel)



Ant Design