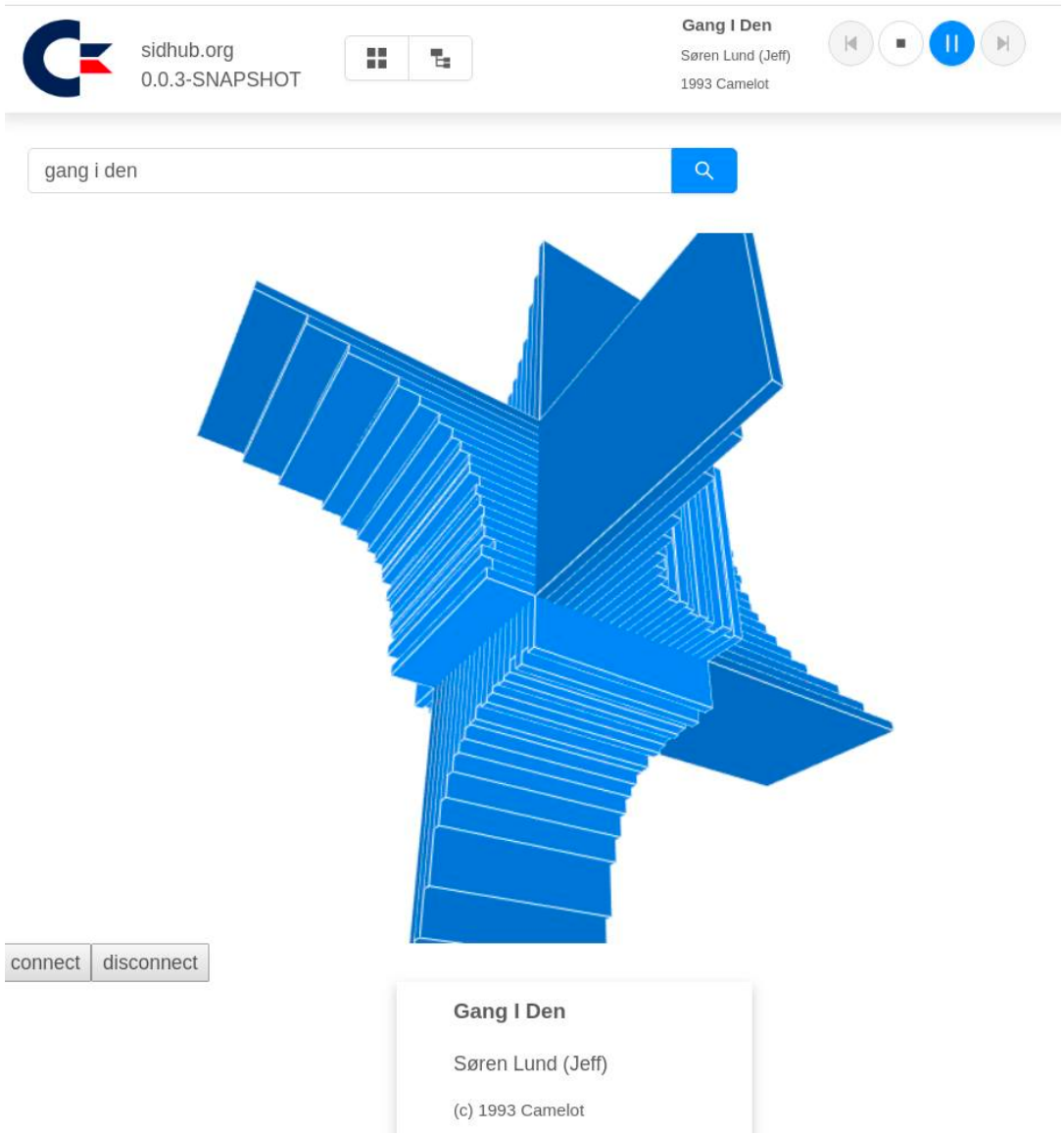
Datenfluss, Kommunikation und Komponentenbau

Marc Teufel



http://www.sidhub.org


hama
®

Rich Thin Clients – bessere GUIs durch neue Clientkonzepte

ASPHeute AspHeut
ASP Aktuell ▶ ASP Welt ▶ ASP German (D

Liste

.NET 2.0 (1)
.NET Allgemein (16)
.NET Fu (5)
ADO.NET (11)
Aprilscherz (3)
ASP Grundlagen (44)
ASP Tricks (83)
ASP.NET (44)
ASPIntranet.de (5)
C# (28)
Datenbank (44)
Dokumentation (4)
IIS 6.0 (1)
Komponenten (29)
Optimierung (10)
Server (21)
Sicherheit (34)
Tee Off (6)
VB.NET (6)
WAP (8)
Web Services (11)
XML (9)

RSS

Suchen

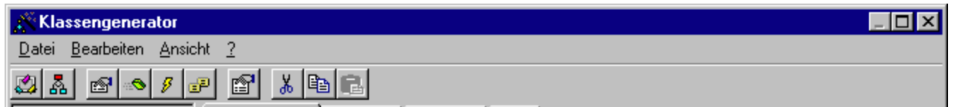
Klassenauflistungen mit dem Dictionary-Objekt

Geschrieben von: Marc Teufel
Kategorie: ASP Tricks
24055 Zugriffe. 25 Bewertungen, Durchschnitt 1.68

Mit Klassenauflistungen hat jeder ernsthafte Visual Basic-Programmierer irgendwann einmal zu tun. Ob freiwillig oder unfreiwi
Verfügung steht. Was also tun? Dieser Artikel zeigt einen kleinen Trick mit dem Klassenauflistungen sehr einfach aufgebaut und

Wie ging das nochmal in Visual Basic?

Klassenauflistungen in Visual Basic sind relativ einfach zu handhaben, da ja dort ein Klassengenerator zur Verfügung steht der d



https://github.com/mteufel/more-vue.git

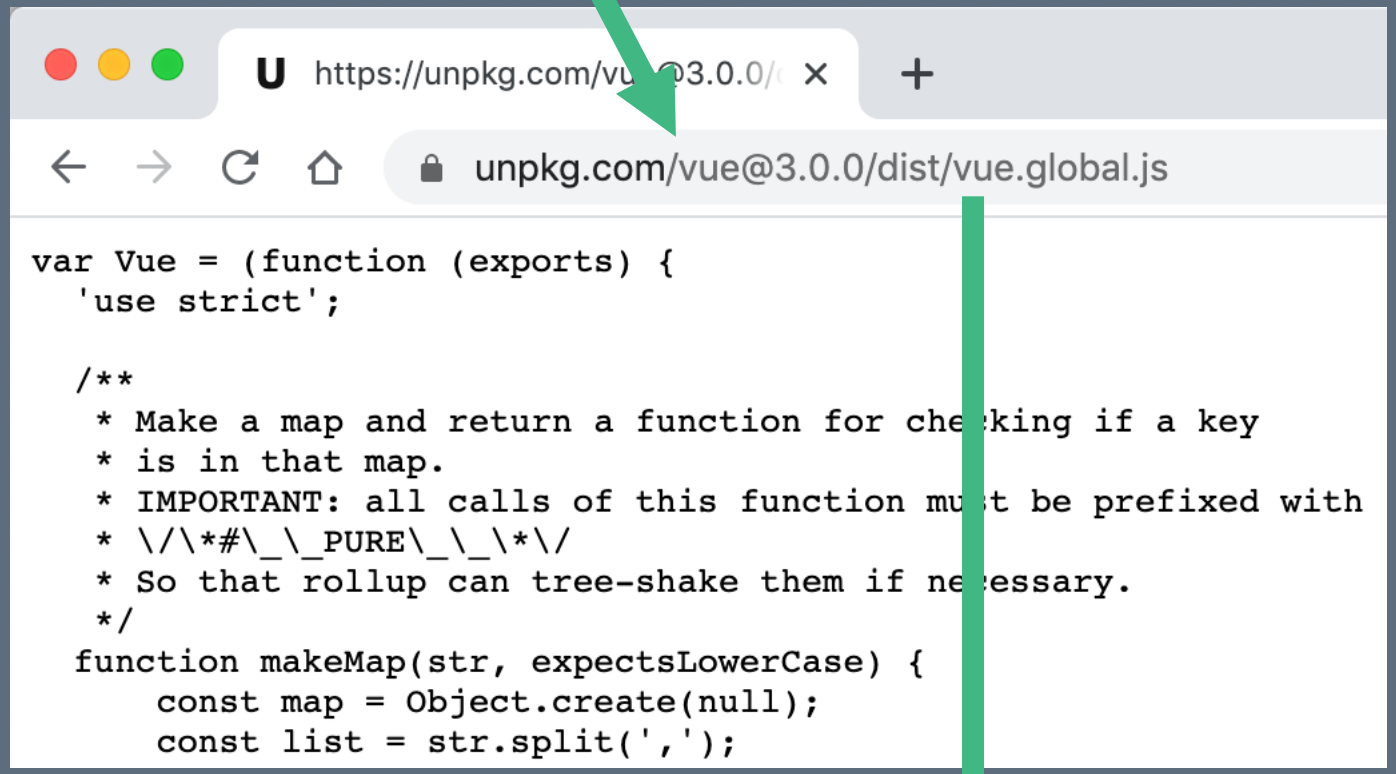
Teil 1

Komponentenbau

easy.html

```
<!doctype html> <html lang="en">
    <head>
        <script src="https://unpkg.com/vue@3.0.0/dist/vue.runtime.global.js"></script>
        <title>More Vue</title>
    </head>
    <body>
        <div id="counter">
            <h1>Counter: {{ counter }}</h1>
        </div>
    </body>
    <script>
        const Counter = {
            data() {
                return { counter: 0 }
            },
            mounted() {
                setInterval(() => { this.counter++ }, 1000)
            }
        }
        Vue.createApp(Counter).mount('#counter')
    </script>
</html>
```

https://unpkg.com/vue@next



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

easy.html

```
<!doctype html> <html lang="en">
    <head>
        <script src="https://unpkg.com/vue@3.0.0/dist/vue.runtime.global.js"></script>
        <title>More Vue</title>
    </head>
    <body>
        <div id="counter">
            <h1>Counter: {{ counter }}</h1>
        </div>
    </body>
    <script>
        const Counter = {
            data() {
                return { counter: 0 }
            },
            mounted() {
                setInterval(() => { this.counter++ }, 1000)
            }
        }
        Vue.createApp(Counter).mount('#counter')
    </script>
</html>
```

Template

Komponente

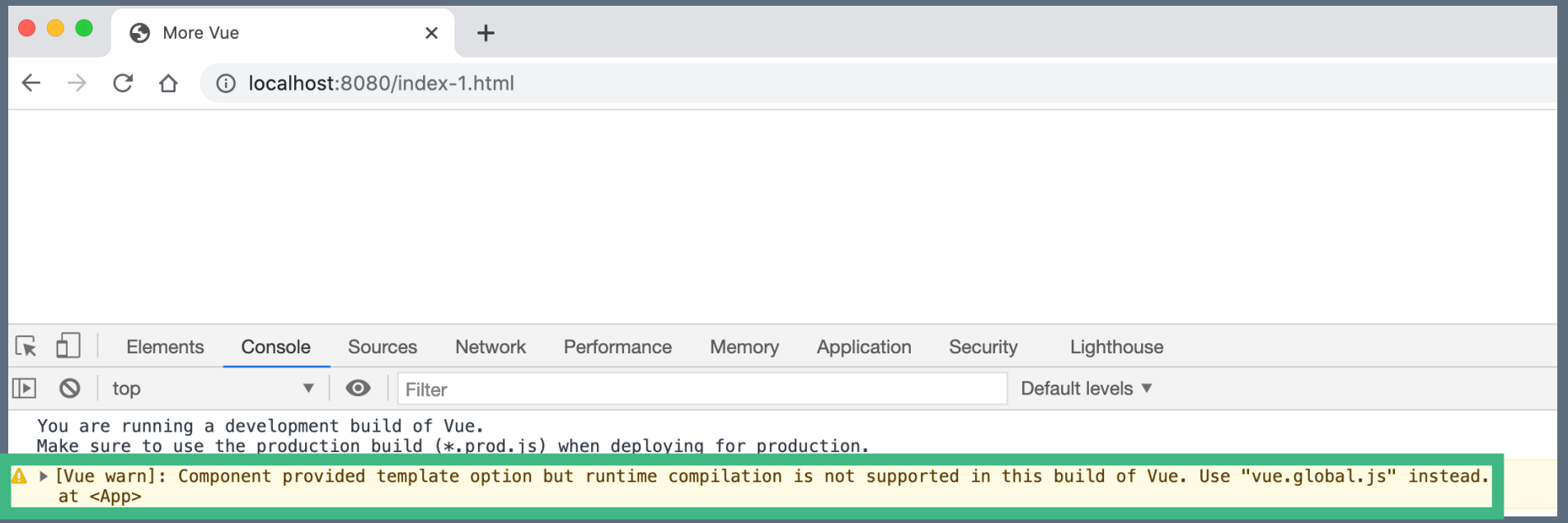
index-1.html

```html
<!doctype html> <html lang="en">
    <head>
        <script src="https://unpkg.com/vue@3.0.0/dist/vue.runtime.global.js"></script>
        <title>More Vue</title>
    </head>
    <body>
        <div id="counter">
            <h1>Counter: {{ counter }}</h1>
        </div>
    </body>
    <script src="counter-1.js"></script>
</html>

```

counter-1.js

```js
const Counter = {
    data() {
        return { counter: 0 }
    },
    mounted() {
        setInterval(() => { this.counter++ }, 1000)
    }
}
Vue.createApp(Counter).mount('#counter')

```

```
marcteufel@mac:~/web-prj/more-vue$ npx servor --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

```
<!doctype html> <html lang="en">
    <head>
        <script src="https://unpkg.com/vue@3.0.0/dist/vue.runtime.global.js"></script>
        <title>More Vue</title>
    </head>
    <body>
        <div id="counter">
            <h1>Counter: {{ counter }}</h1>
        </div>
    </body>
    <script src="counter-1.js"></script>
</html>

```

Inline Templating!

counter-1.js

```
const Counter = {
    data() {
        return { counter: 0 }
    },
    mounted() {
        setInterval(() => { this.counter++ }, 1000)
    }
}
Vue.createApp(Counter).mount('#counter')

```

index-1.html

```html
<!doctype html> <html lang="en">
    <head>
        <script src="https://unpkg.com/vue@3.0.0/dist/vue.global.js"></script>
        <title>More Vue</title>
    </head>
    <body>
        <div id="counter">
            <h1>Counter: {{ counter }}</h1>
        </div>
    </body>
    <script src="counter-1.js"></script>
</html>
```

counter-1.js

```js
const Counter = {


    data() {
        return { counter: 0 }
    },
    mounted() {
        setInterval(() => { this.counter++ }, 1000)
    }
}

Vue.createApp(Counter).mount('#counter')
```

index-2.html

```html
<!doctype html> <html lang="en">
    <head>
        <script src="https://unpkg.com/vue@3.0.0/dist/vue.global.js"></script>
        <title>More Vue</title>
    </head>
    <body>
        <div id="counter"></div>
    </body>
    <script src="counter-2.js"></script>
</html>
```

counter-2.js

```js
const Counter = {

    template: '<h1>Counter: {{ counter }}</h1>',

    data() {
        return { counter: 0 }
    },
    mounted() {
        setInterval(() => { this.counter++ }, 1000)
    }
}
Vue.createApp(Counter).mount('#counter')
```

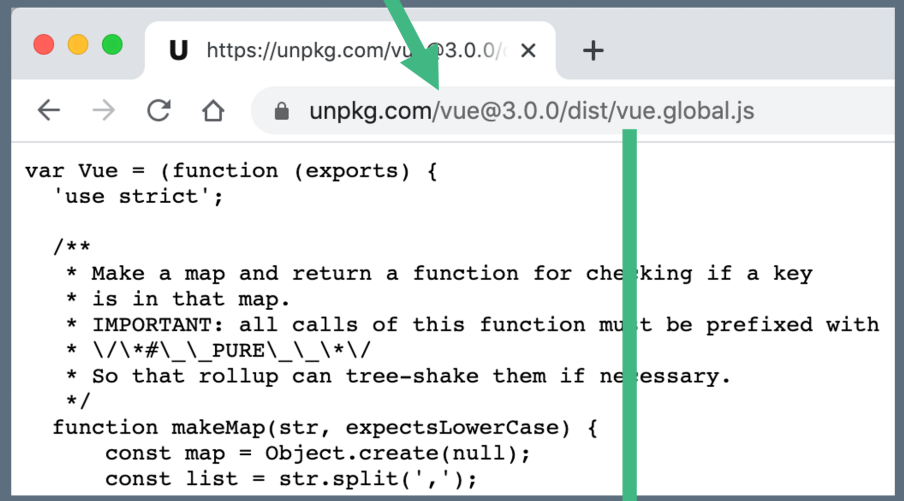
String `Template Literals`

ESM

EcmaScript Modules

verwenden!

https://unpkg.com/vue@next



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

ohne

index-2.html

```html
<!doctype html> <html lang="en">
    <head>
        <script src="https://unpkg.com/vue@3.0.0/dist/vue.global.js"></script>
        <title>More Vue</title>
    </head>
    <body>
        <div id="counter"></div>
    </body>
    <script src="counter-2.js"></script>
</html>
```

counter-2.js

```js
const Counter = {

    template: '<h1>Counter: {{ counter }}</h1>',

    data() {
        return { counter: 0 }
    },
    mounted() {
        setInterval(() => { this.counter++ }, 1000)
    }
}
Vue.createApp(Counter).mount('#counter')
```

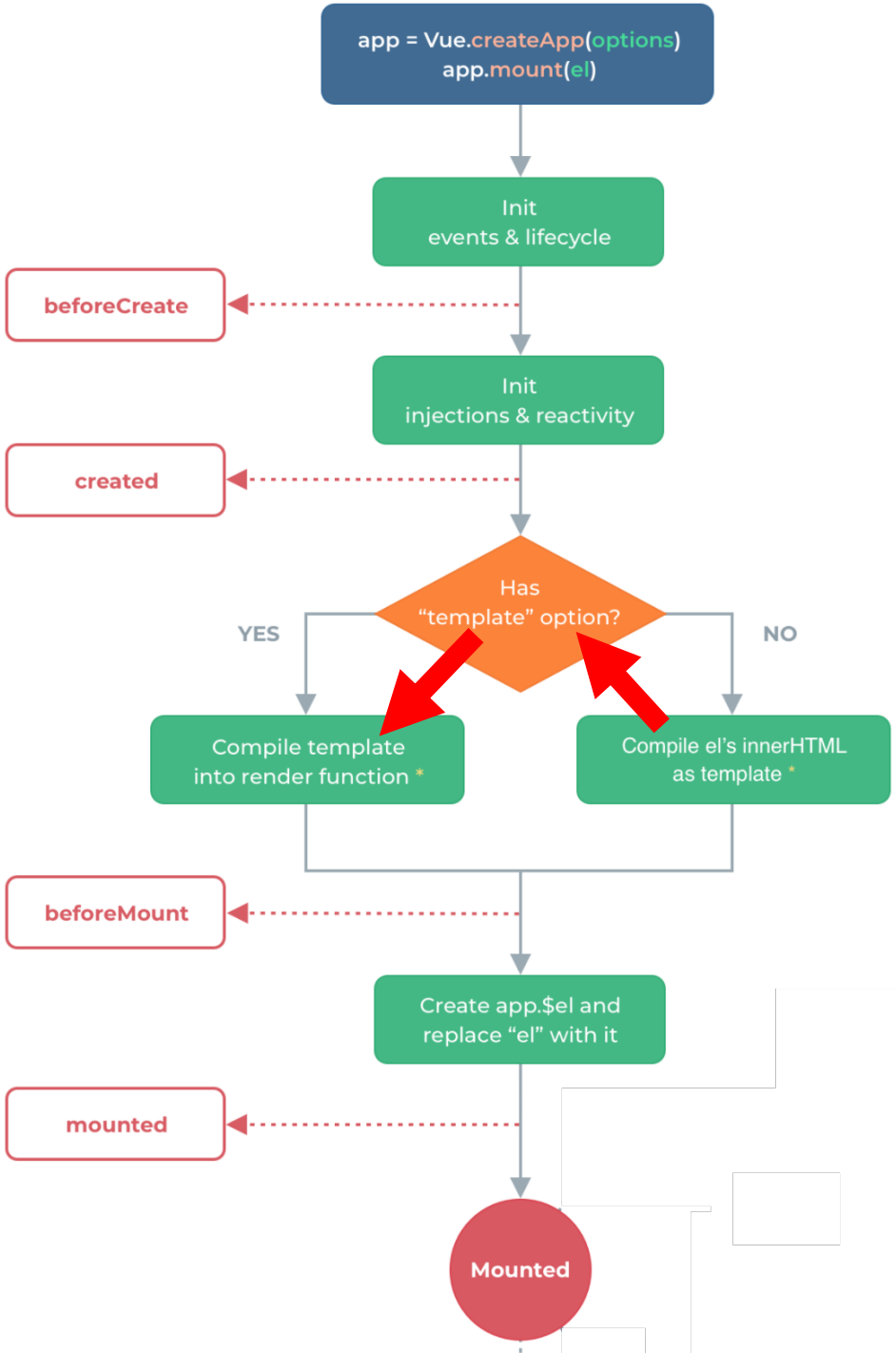
mit

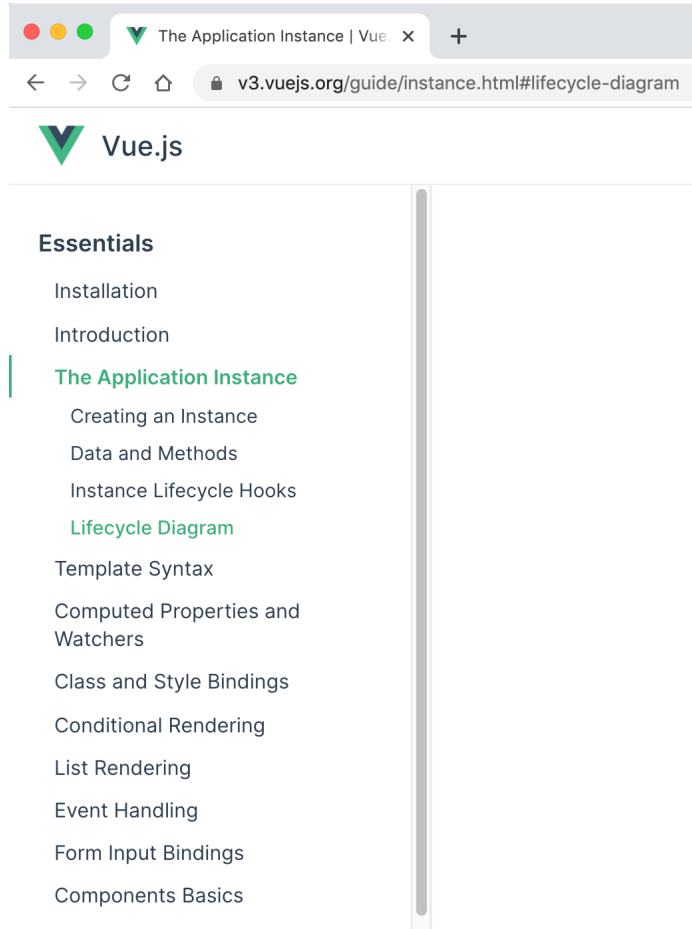
index-esm.html

```html
<!doctype html> <html lang="en">
    <head>
        <script src="counter-esm.js" type="module"></script>
        <title>More Vue</title>
    </head>
    <body>
        <div id="counter"></div>
    </body>
</html>
```

counter-esm.js

```js
import { createApp }
        from 'https://unpkg.com/vue@3.0.0-rc.7/dist/vue.esm-browser.js'

const Counter = {

    template: '<h1>Counter: {{ counter }}</h1>',

    data() {
        return { counter: 0 }
    },
    mounted() {
        setInterval(() => { this.counter++ }, 1000)
    }
}
createApp(Counter).mount('#counter')
```


app = Vue.createApp(options)
app.mount(el)
Init
events & lifecycle
beforeCreate
Init
injections & reactivity
created
Has
“template” option?
YES
NO
Compile template
into render function *
Compile el’s innerHTML
as template *
beforeMount
Create app.$el and
replace “el” with it
mounted
Mounted


The Application Instance | Vue
v3.vuejs.org/guide/instance.html#lifecycle-diagram
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

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

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

Elements Console Sources Network

```
<!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>
```

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>
15
```

Tipp:
So lieber nicht!

Wiederverwendbare Komponenten bauen

component.html

```html
<!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>
```

component.js

```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*

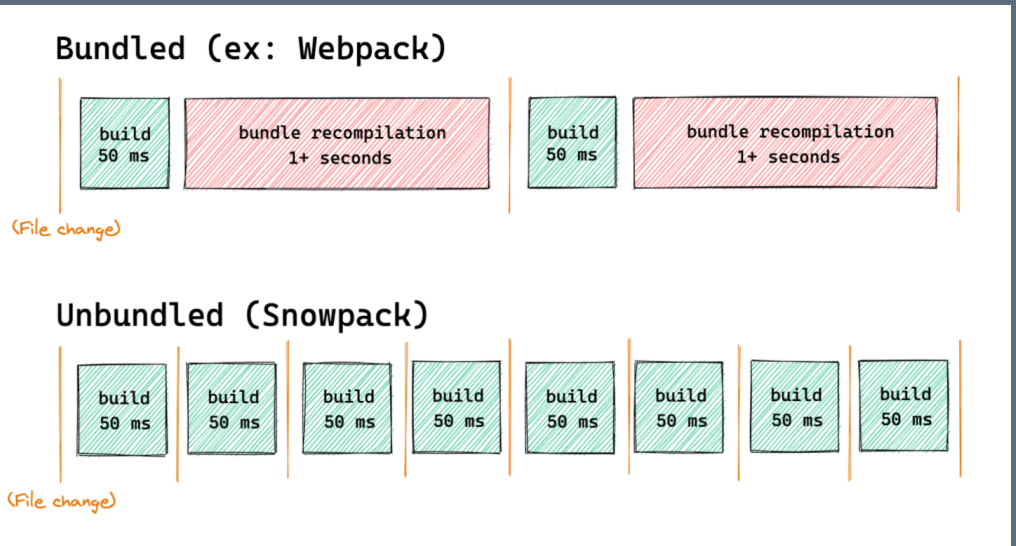
ab Vue 3, Entwicklung ohne Bundler auf Basis von ESM:

- 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



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.

https://github.com/mteufel/more-vue.git

Teil 2

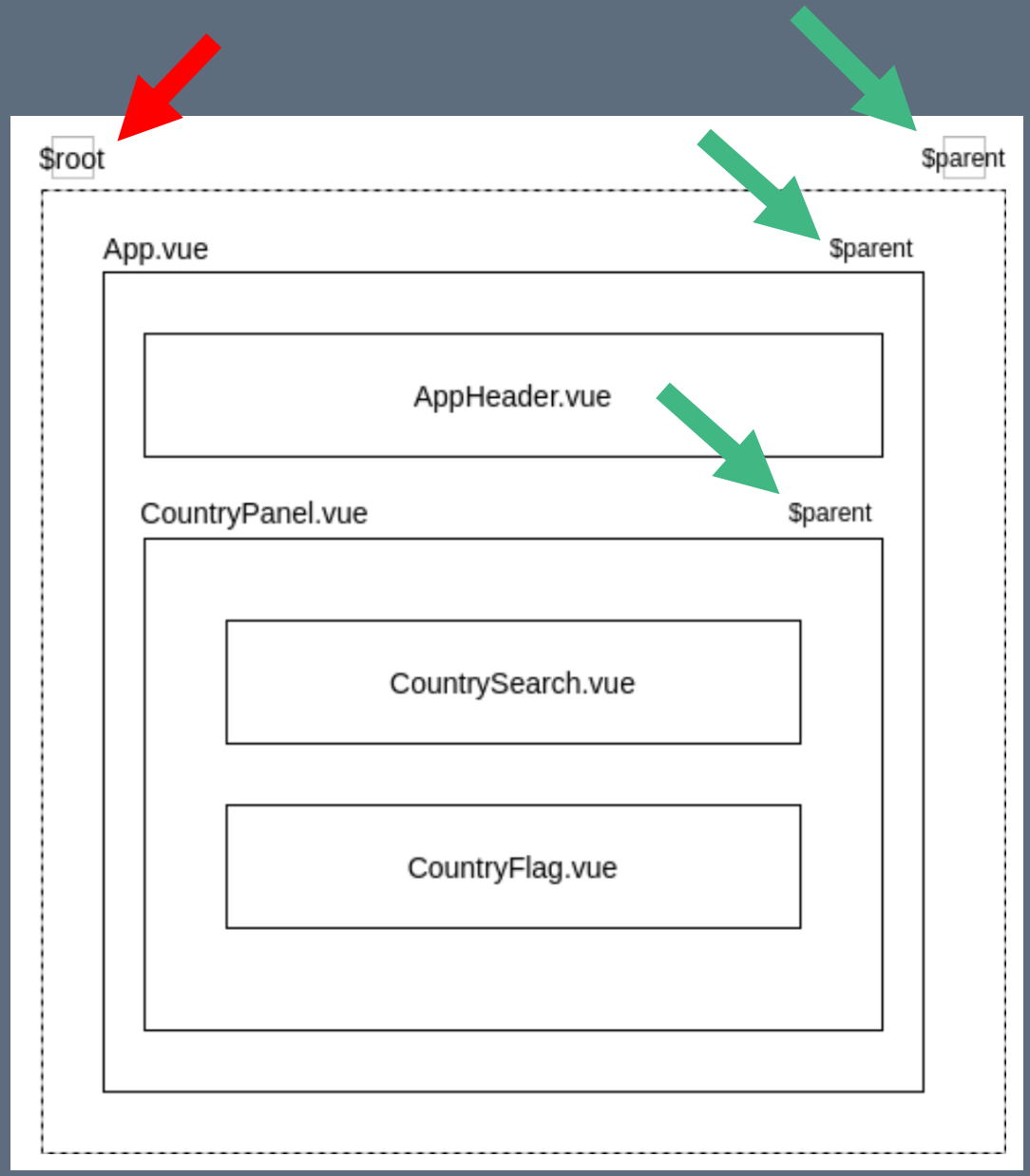
Datenfluss, Kommunikation

Aller guten Dinge sind DREI!

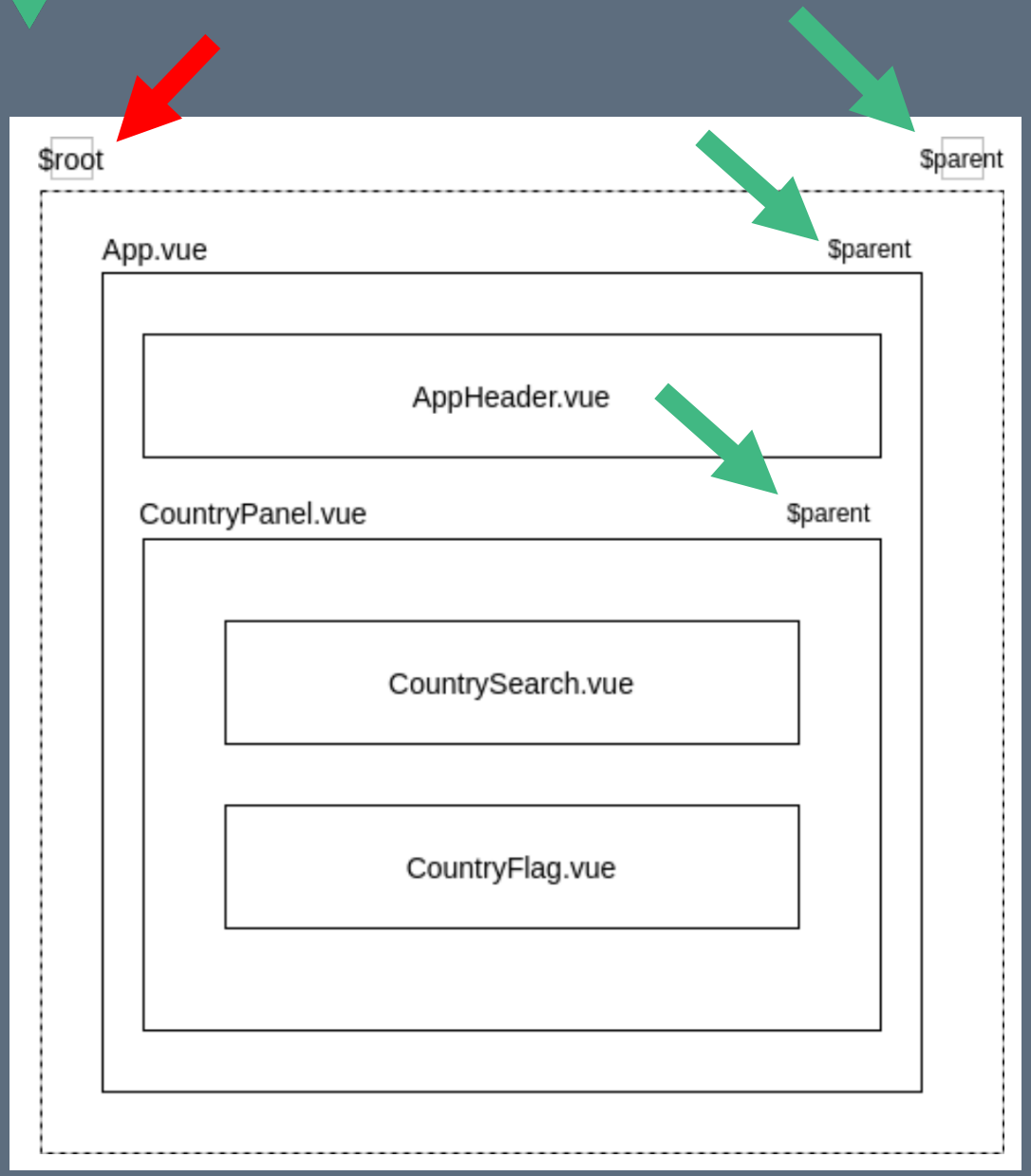
- $root/$parent

- Durchreichen → Props/Emit
- Rumreichen → Globale Stores, State Management
- Reinreichen → Provide/Inject

$root/$parent


$root
$parent
App.vue
$parent
AppHeader.vue
CountryPanel.vue
$parent
CountrySearch.vue
CountryFlag.vue

$root/$parent

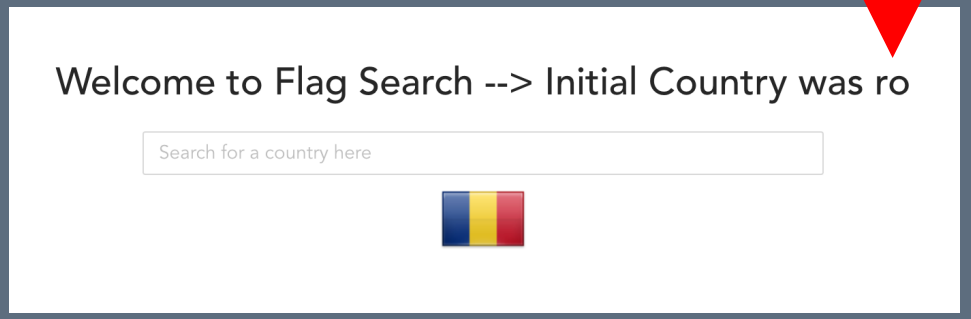


main.js

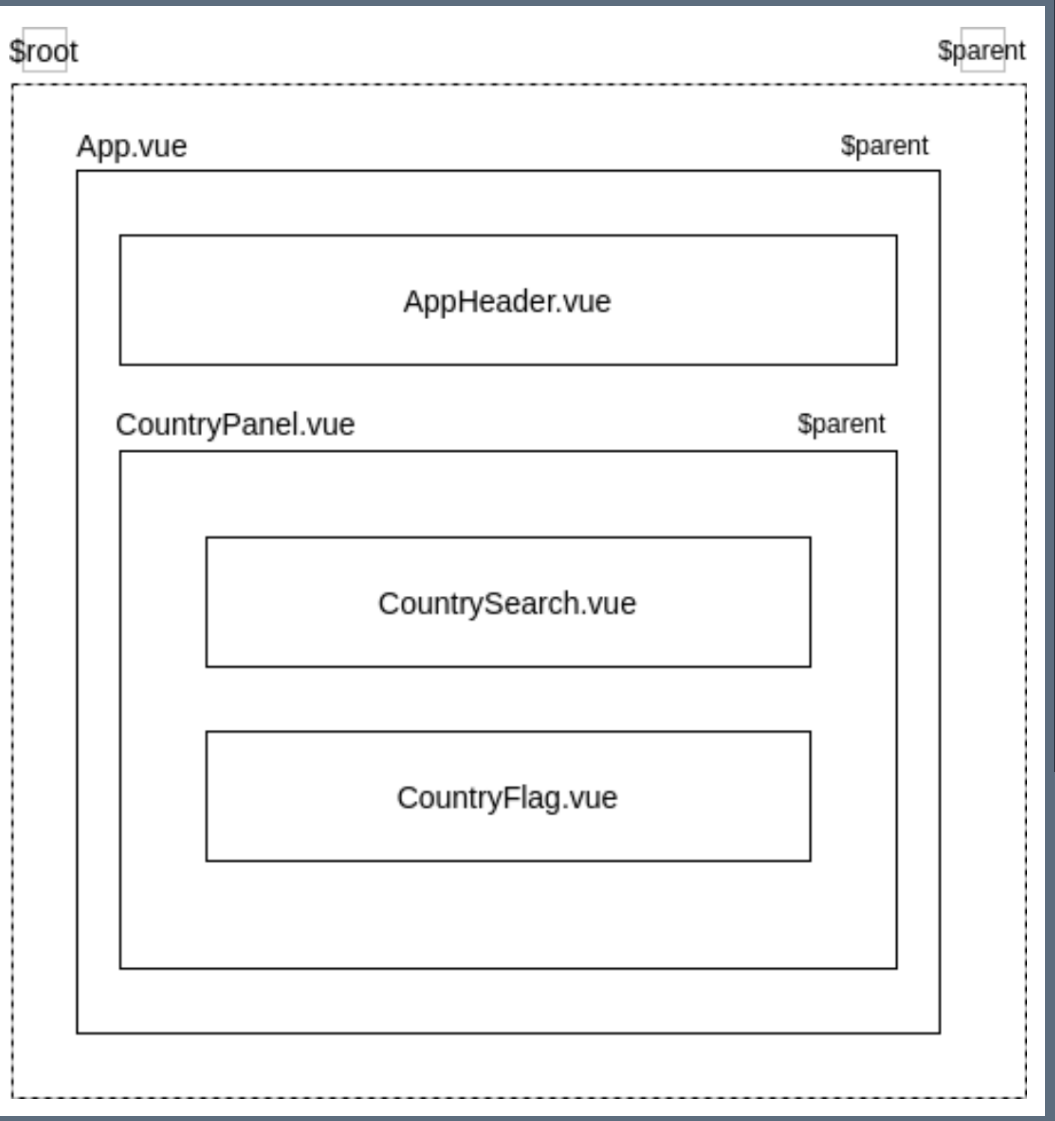
```
import { createApp, h } from 'vue'
import Antd from 'ant-design-vue'
import App from './App.vue'
import 'ant-design-vue/dist/antd.css'


const rootInstance = createApp( {
    data() {
      return { initialCountry: 'ro' }
    },
    render() {
      return h(App)
    }
  }
)
rootInstance.config.productionTip = false;
rootInstance.use(Antd)
rootInstance.mount('#app')
```

AppHeader.vue

```
<template>
  <div>
    <h1>{{ title }} --> Initial Country was {{ this.$root.initialCountry }}</h1>
  </div>
</template>

<script>
export default {
  name: "AppHeader",
  props: {
    title: String
  }
};
</script>
```



$root/$parent

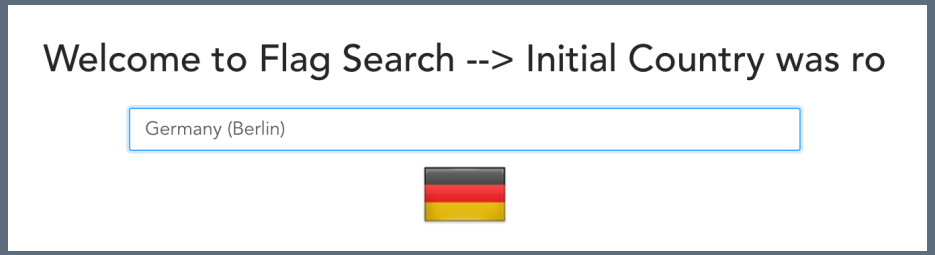


App.vue

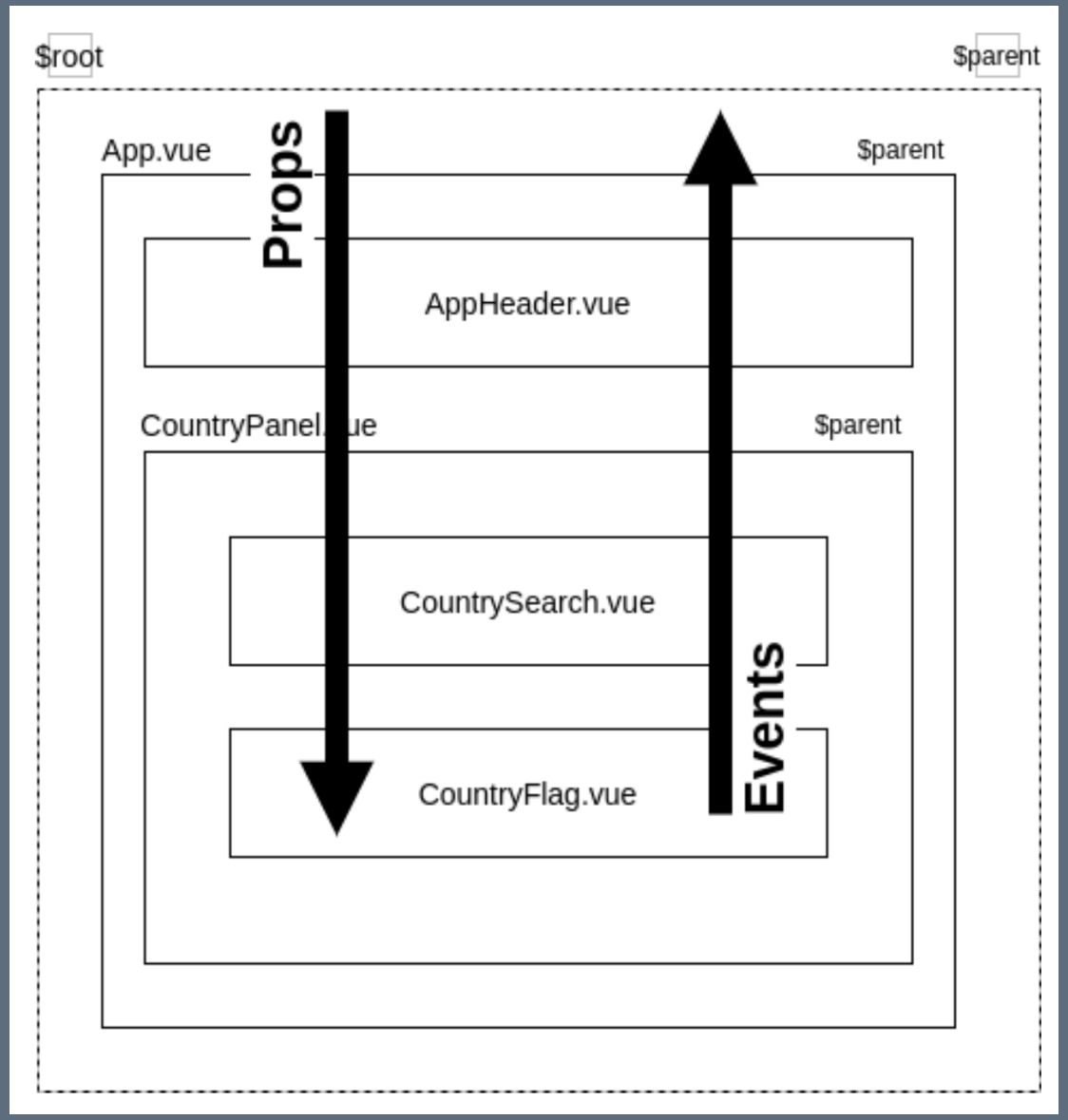
```vue
<template>
  <div id="app">
    <AppHeader title="Welcome to Flag Search" />
    <CountryPanel>
      <CountrySearch />
      <CountryFlag :country="country" />
    </CountryPanel>
  </div>
</template>

<script>
import AppHeader from "./components/AppHeac
import CountrySearch from "./components/Cou
import CountryPanel from "./components/Cour
import CountryFlag from "./components/Count


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

CountrySearch.vue

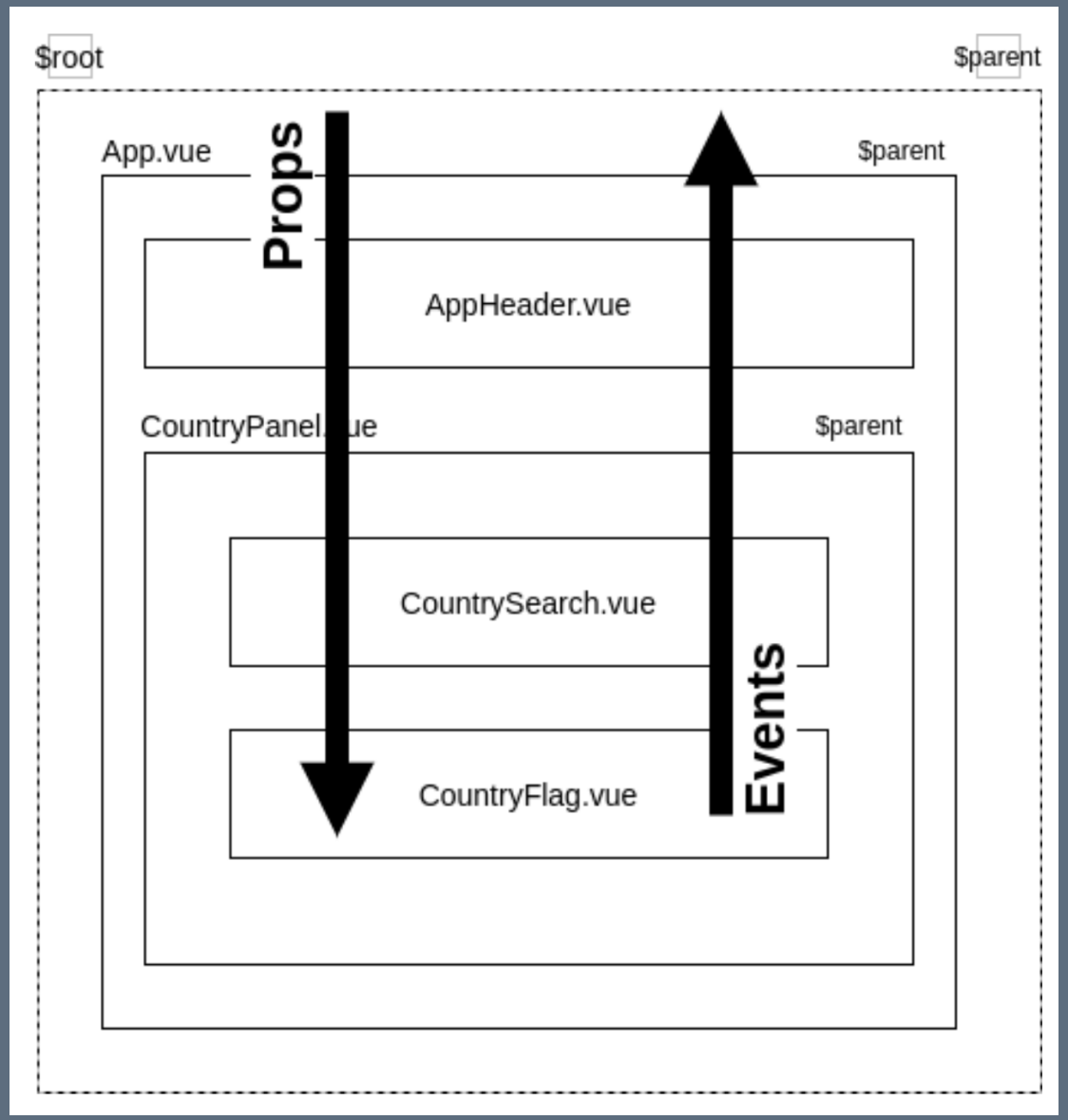
```vue
<template>
  <div>
    <a-select
      show-search
      placeholder="Search for a country here"
      style="width: 500px"
      :filter-option="false"
      :default-active-first-option="false"
      :show-arrow="false"
      @search="onSearch"
      @change="onSelection">
      <a-select-option v-for="o in options" :key="o.value">
        {{ o.text }}
      </a-select-option>
    </a-select>
  </div>
</template>

<script>

import { useSearch } from '../use/useCountries'
export default {
  name: "CountrySearch",
  setup() {
    return { ...useSearch() }

  },
  methods: {
    onSelection(selectedValue) {
      this.$parent.$parent.country = selectedValue;
    }
  }
}
</script>
```



Daten durchreichen mit dem Standard-Weg: Props/Emit



Daten durchreichen mit dem Standard-Weg: Props/Emit

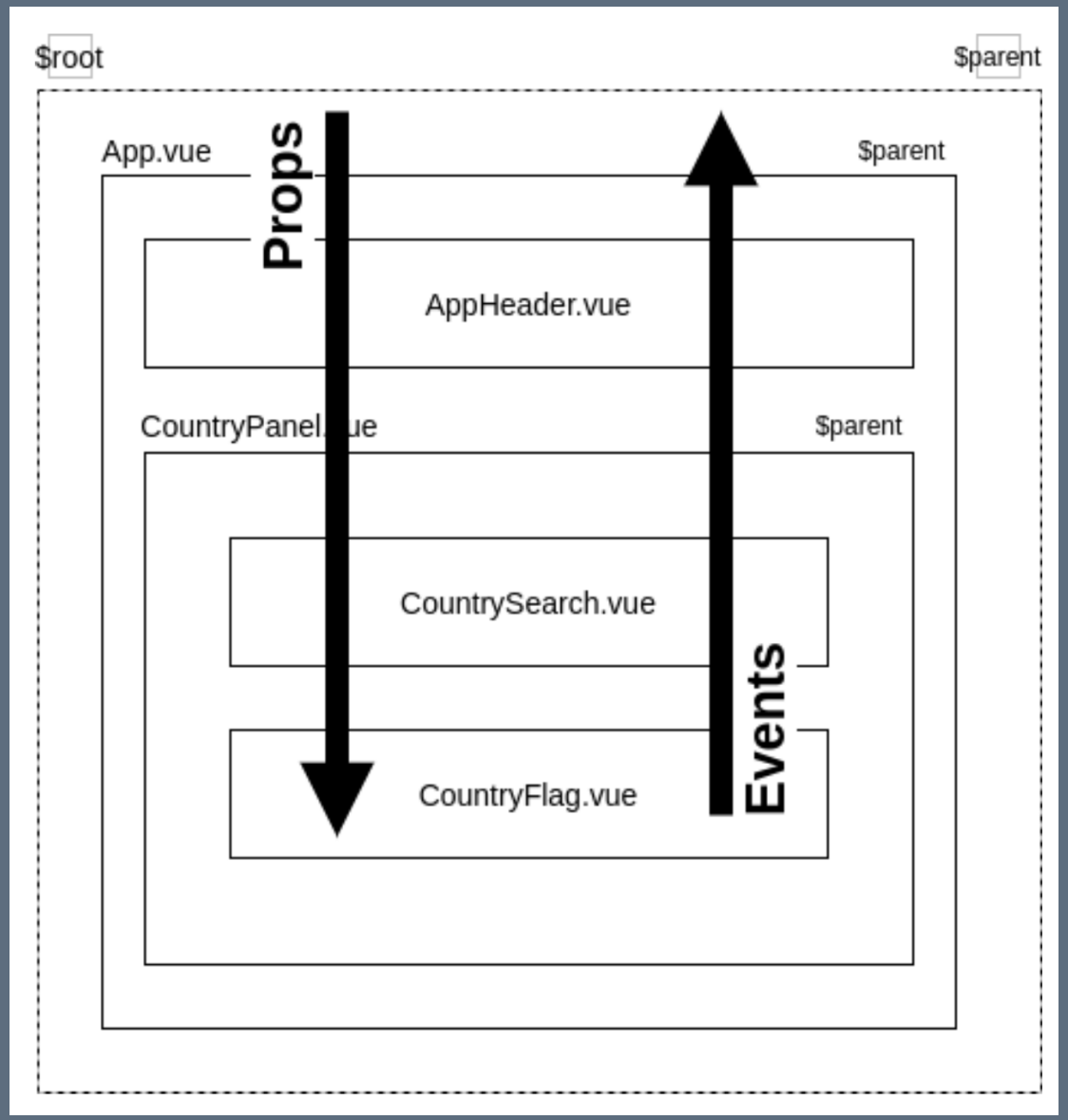


App.vue

You, a few seconds ago | 1 author (You)

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

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


export default {
  name: "App",
  data() {
    return {
      country: 'gb'
    }
  },
  methods: {
    countryChanged(selectedCountry) {
      console.log('event angekommen: ' + selectedCountry)
      this.country = selectedCountry
    }
  },
  components: {
    AppHeader,
    CountryPanel,
    CountrySearch,
    CountryFlag
  }
};
</script>
```

Daten durchreichen mit dem Standard-Weg: Props/Emit



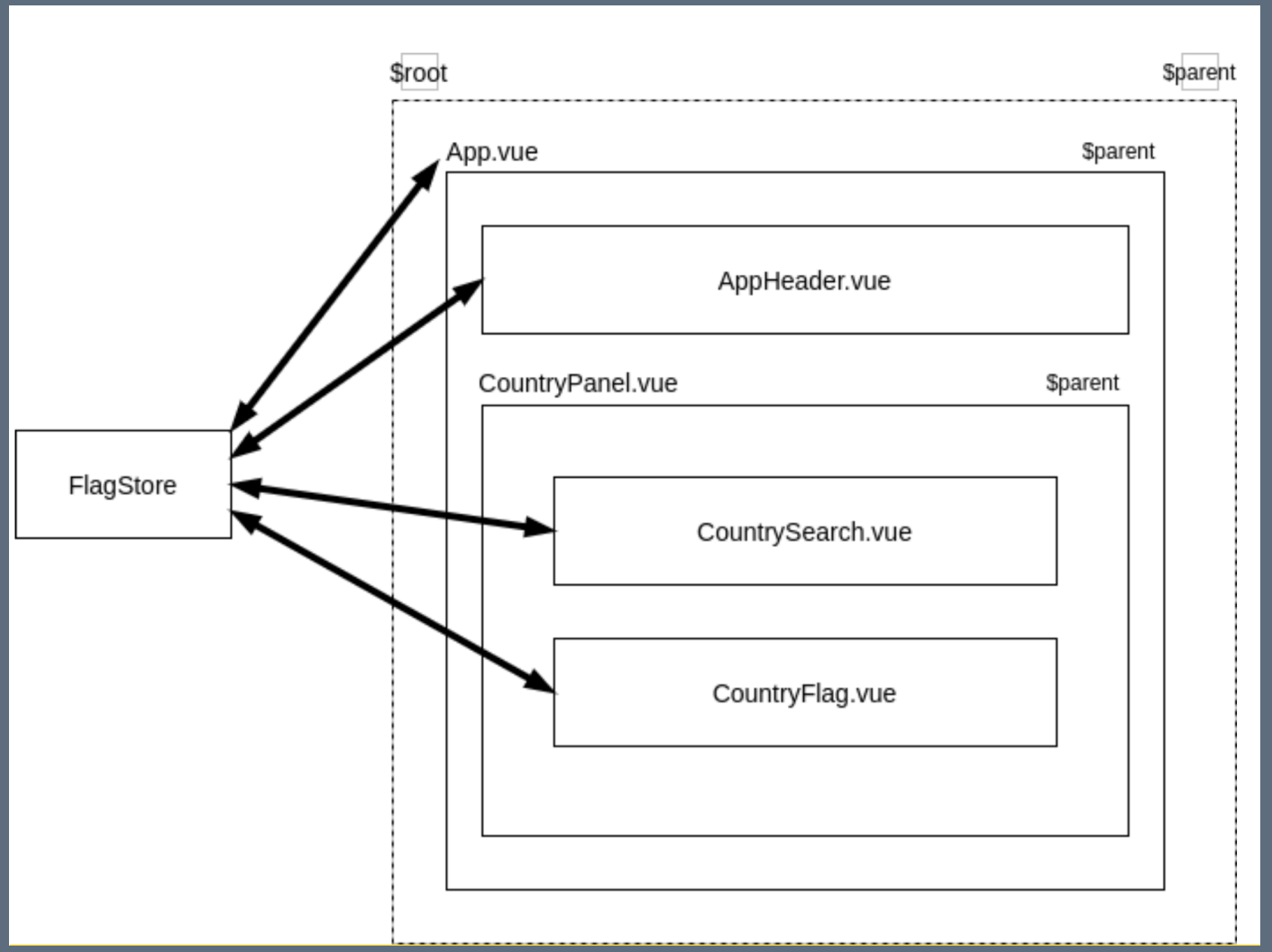
App.vue

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

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


export default {
  name: "App",
  data() {
    return {
      country: 'gb'
    }
  },
  methods: {
    countryChanged(selectedCount
      console.log('event angekom
      this.country = selectedCou
    }
  },
  components: {
    AppHeader,
    CountryPanel,
    CountrySearch,
    CountryFlag
  }
};
</script>
```

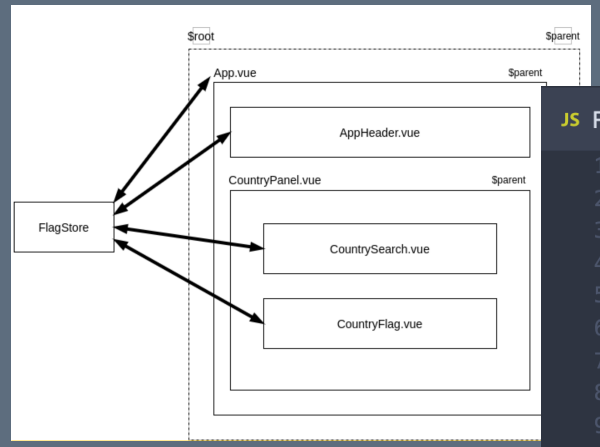
CountrySearch.vue

```vue
          :default-active-first-option="false"
          :show-arrow="false"
          @search="onSearch"
          @change="onSelection">
          <a-select-option v-for="o in options" :key="o.value">
            {{ o.text }}
          </a-select-option>
      </a-select>
    </div>
</template>

<script>

import { useSearch } from '../use/useCountries'
export default {
  name: "CountrySearch",
  setup() {
    return { ...useSearch() }

  },
  methods: {
    onSelection(selectedValue) {
      this.$emit('country-changed', selectedValue)
    }
  }
}
</script>
```

Daten rumreichen mit Stores



Daten rumreichen mit Stores



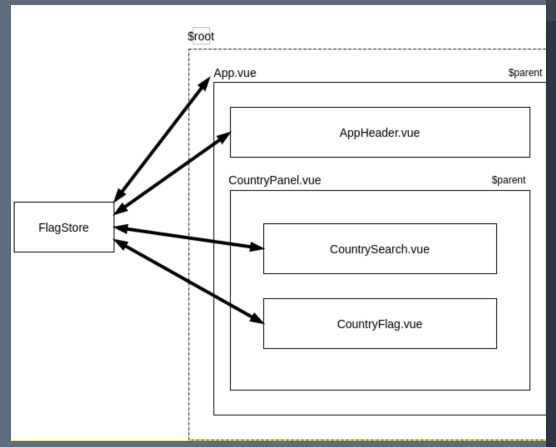
FlagStore.js

```
const createFlagStore = () => {

    const url = 'https://restcountries.eu/rest/v2'

    const queryByName = async (search) => {
        let response = await fetch(`${url}/name/` + search)
        return await response.json()
    }

    let options = []
    let selectedCountry = 'it'

    return {

        getOptions: () => options,
        getSelectedCountry: () => selectedCountry,

        doSearch: async(search) => {
            if (search === "") {
                return
            }
            let json = await queryByName(search)
            options = await json.map(e => {
                return { text: `${e.name} (${e.capital})`,
                         value: e.alpha2Code.toLowerCase() }
            })
        }

    }
}

const flagStore = createFlagStore()
export default flagStore
```

CountrySearch.vue

```
            :filter-option="false"
            :default-active-first-option="false"
            :show-arrow="false"
            @search="onSearch"
            @change="onSelection">
            <a-select-option v-for="o in options" :key="o.value">
              {{ o.text }}
            </a-select-option>
        </a-select>
    </div>
</template>

<script>

import { ref } from 'vue'
import flagStore from '../stores/FlagStore'

export default {
  name: "CountrySearch",
  setup() {
    const options = ref([])
    return { options }
  },
  methods: {
    onSearch(search) {
      flagStore.doSearch(search)
      this.options = flagStore.getOptions();
    },
    onSelection(selectedValue) {
      // TODO
    }
  }
}
</script>
```

Observer Pattern hilft!



FlagStore.js

```js
const createFlagStore = () => {

    const url = 'https://restcountries.eu/rest/v2'

    const queryByName = async (search) => {
        let response = await fetch(`${url}/name/` + search)
        return await response.json()
    }

    let options = []
    let selectedCountry = 'it'
    let subscribers = []

    return {

        getOptions: () => options,
        getSelectedCountry: () => selectedCountry,

        doSearch: async(search) => {
            if (search === "") {
                return
            }
            let json = await queryByName(search)
            options = await json.map(e => {
                return { text: `${e.name} (${e.capital})`,
                         value: e.alpha2Code.toLowerCase() }
            })
        },

        subscribe: (fn) => {
            subscribers.push(fn)
        },
        countrySelected: (sel) => {
            selectedCountry = sel;
            subscribers.forEach( s =>s())
        }

    }
}

const flagStore = createFlagStore()
export default flagStore
```

App.vue

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

<script>

import AppHeader from "./components/AppHeader.vue"
import CountrySearch from "./components/CountrySearch.vue"
import CountryPanel from "./components/CountryPanel.vue"
import CountryFlag from "./components/CountryFlag.vue"
import flagStore from "./stores/FlagStore"
import { ref } from 'vue'

export default {
  name: "App",
  setup() {

    const country = ref(flagStore.getSelectedCountry())

    flagStore.subscribe( () => country.value = flagStore.getSelectedCountry() )

    return { country}
  },
  components: {
    AppHeader,
    CountryPanel,
    CountrySearch,
    CountryFlag
  }
};
</script>
```

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

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

<script>

import AppHeader from "./components/AppHeader.vue"
import CountrySearch from "./components/CountrySearch.vue"
import CountryPanel from "./components/CountryPanel.vue"
import CountryFlag from "./components/CountryFlag.vue"
import { ref, provide } from 'vue'

export default {
  name: "App",
  setup() {

    const country = ref('us')
    provide('country', country)
    return { country }

  },
  components: {
    AppHeader,
    CountryPanel,
    CountrySearch,
    CountryFlag
  }
};
</script>
```

Dependency Injection mit Provide/Inject

App.vue

```vue
<template>
  <div id="app">
    <AppHeader title="Welcome to Fla
    <CountryPanel>
      <CountrySearch />
      <CountryFlag />
    </CountryPanel>
  </div>
</template>

<script>

import AppHeader from "./components/
import CountrySearch from "./compone
import CountryPanel from "./componen
import CountryFlag from "./component
import { ref, provide } from 'vue'

export default {
  name: "App",
  setup() {

    const country = ref('us')
    provide('country', country)
    return { country }

  },
  components: {
    AppHeader,
    CountryPanel,
    CountrySearch,
    CountryFlag
  }
};
</script>
```

CountrySearch.vue

```vue
        show-search
        placeholder="Search for a country here"
        style="width: 500px"
        :filter-option="false"
        :default-active-first-option="false"
        :show-arrow="false"
        @search="onSearch"
        @change="onSelection">
        <a-select-option v-for="o in options" :key="o.value">
          {{ o.text }}
        </a-select-option>
      </a-select>
    </div>
</template>

<script>

import { useSearch } from '../use/useCountries'
import { inject } from 'vue'


export default {
  name: "CountrySearch",
  setup() {
    const country = inject('country')
    return { country,
             ...useSearch() }
  },
  methods: {
    onSelection(selectedValue) {
      this.country = selectedValue
    }
  }
}
</script>
```

Dependency Injection mit Provide/Inject

App.vue

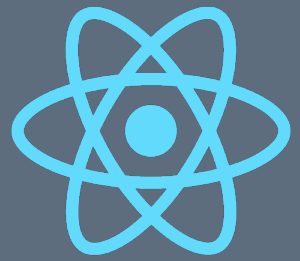
```
<template>
  <div id="app">
    <AppHeader title="Welcome to Flag Search" />
    <CountryPanel>
      <CountrySearch />
      <CountryFlag />
    </CountryPanel>
  </div>
</template>

<script>

import AppHeader from "./components/AppHeader.vue"
import CountrySearch from "./components/CountrySearch.vue"
import CountryPanel from "./components/CountryPanel.vue"
import CountryFlag from "./components/CountryFlag.vue"
import { ref, provide } from 'vue'

export default {
  name: "App",
  setup() {

    const country = ref('us')
    provide('country', country)
    return { country }

  },
  components: {
    AppHeader,
    CountryPanel,
    CountrySearch,
    CountryFlag
  }
};
</script>
```

CountrySearch.vue

```
            show-search
            placeholder="Search for a country here"
            style="width: 500px"
            :filter-option="false"
            :default-active-first-option="false"
            :show-arrow="false"
            @search="onSearch"
            @change="onSelection">
            <a-select-option v-for=
              {{ o.text }}
            </a-select-option>
          </a-select>
        </div>
      </template>

      <script>

      import { useSearch } from '../
      import { inject } from 'vue'


      export default {
        name: "CountrySearch",
        setup() {
          const country = inject('cou
          return { country,
                   ...useSearch() }
        },
        methods: {
          onSelection(selectedValue)
            this.country = selectedVa
          }
        }
      }
      </script>
```

CountryFlag.vue

```
<template>
  <div>
    <img id="flag-image" border="0" src="../assets/blank.gif" v-bind:class="flagCss" />
  </div>
</template>

<script>
import { computed, inject } from 'vue'

export default {
  name: "Flag",
  props: {
    country: String
  },
  setup(props) {

    //const flagCss = computed( () => "flag flag-" + props.country)

    const flagCss = computed( () => "flag flag-" + inject('country').value)

    return { flagCss }
  }
};
</script>
```



ES

Ant Design

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