.msg .consulting .solutions .partnership

Vue.js
Developer friendly, Fast and Versatile

Alexander Schwartz, Principal IT Consultant

Jfokus 2018 – Stockholm (SE) February 2018

Vue.js – developer friendly, fast and versatile

1 Expectations – Tools, Ecosystem, Options

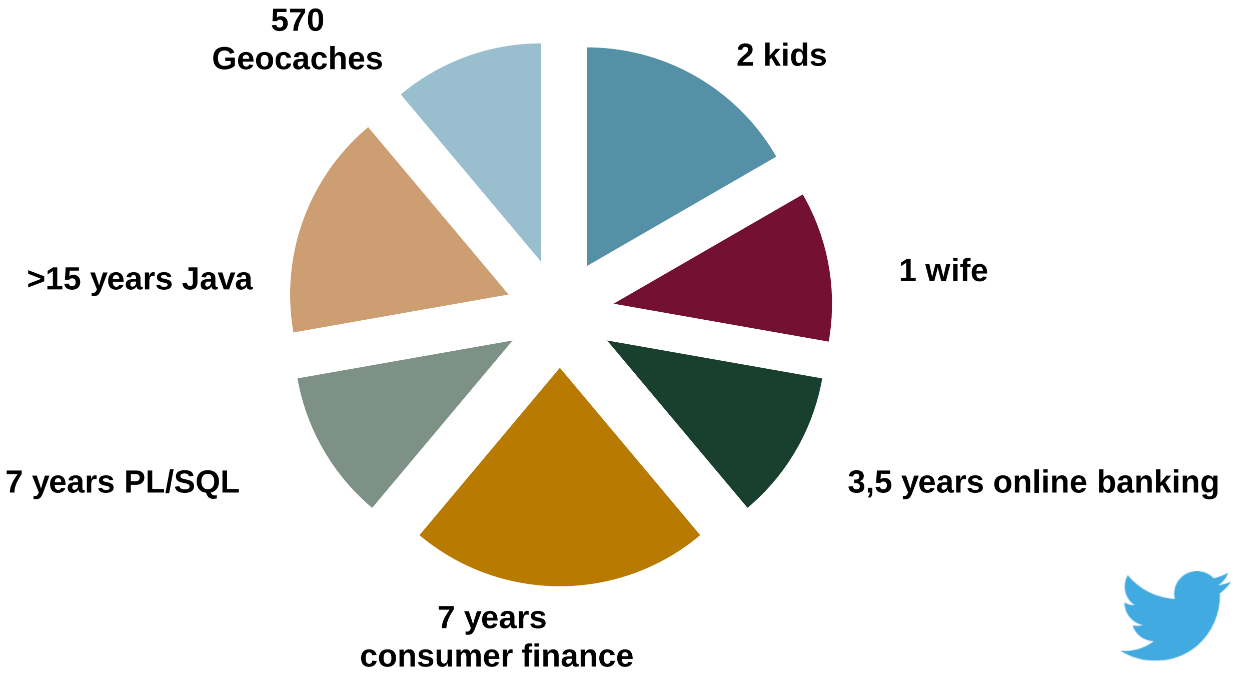
2 Basics – Getting Started

3 Projects – Components and Structure

4 Performance – From different Angles

5 Lessons Learned – Looking back

About me – Principal IT Consultant @ msg Travel & Logistics



@ahus1de

Vue.js – developer friendly, fast and versatile

1 **Expectations – Tools, Ecosystem, Options**

2 Basics – Getting Started

3 Projects – Components and Structure

4 Performance – From different Angles

5 Lessons Learned – Looking back

Why boring is good for your health

JavaScript Fatigue Fatigue:

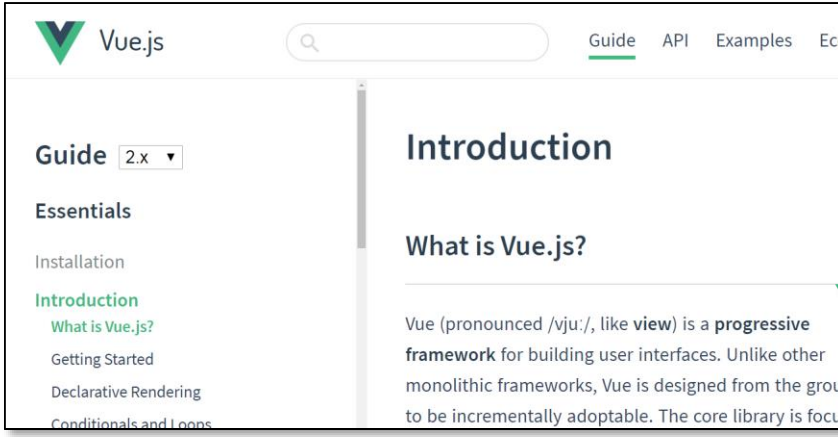
- Too many JavaScript Frameworks
- JavaScript needed to deliver expected User Experience
- Life of the application is longer than the hype cycle of the framework or framework version

What we looked for and found with Vue.js:

- Easy to learn Framework – even for JavaScript newbies
- Productivity in day-to-day work
- No rough corners and edges that lead to problems in the project
- Stable APIs

Getting started

Online-Documentation



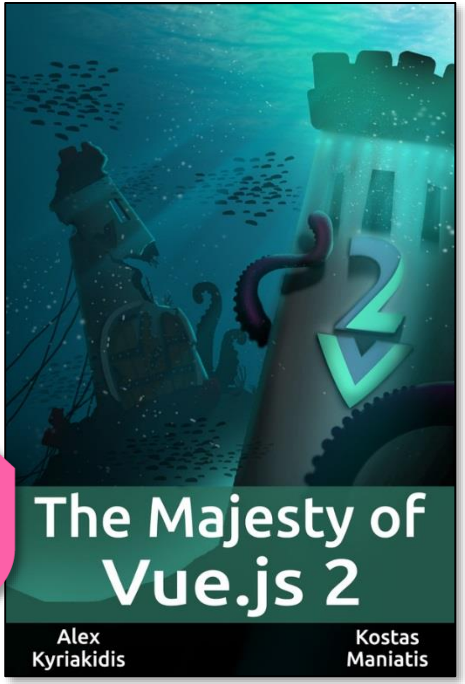
Introduction

What is Vue.js?

Vue (pronounced /vjuː/, like **view**) is a **progressive framework** for building user interfaces. Unlike other monolithic frameworks, Vue is designed from the ground up to be incrementally adoptable. The core library is focused

Awesome
Vue.js Liste

Book (Leanpub)



1. https://github.com/vuejs/awesome-vue
2. https://vuejs.org/v2/guide/
3. https://leanpub.com/vuejs2

The Tools

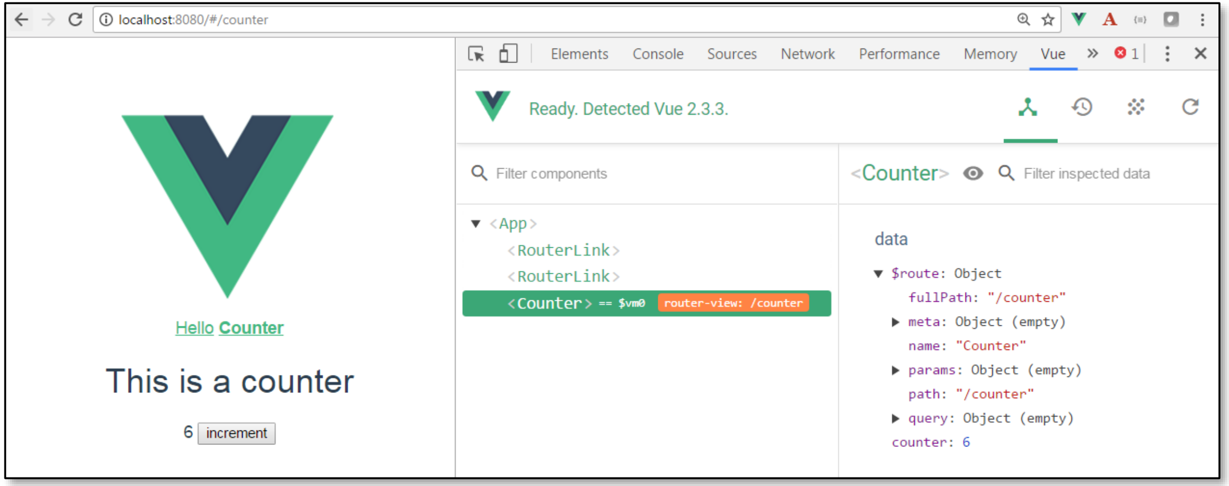
vue-cli: Starter-Project with a complete build chain

- webpack vs. browserify
- simple vs. full feature (hot-reload, linting, testing, progressive web app)

Plugins:

- For IDEs
  use Vue.js plugins by JetBrains
  for IntelliJ/PHPStorm
  or Vetur plugin for VS Code

- For Chrome: Vue.js devtools



Ready. Detected Vue 2.3.3.

This is a counter

6

```
▼ <App>
    <RouterLink>
    <RouterLink>
    <Counter> == $vm0   router-view: /counter
```

<Counter>

data

```
▼ $route: Object
    fullPath: "/counter"
  ▶ meta: Object (empty)
    name: "Counter"
  ▶ params: Object (empty)
    path: "/counter"
  ▶ query: Object (empty)
  counter: 6
```

1. https://github.com/vuejs/vue-cli
2. https://plugins.jetbrains.com/plugin/9442-vue-js
3. https://github.com/vuejs/vue-devtools

Core Functionality and Extensions

Core Functionalities (abbreviated):

- View-, Input-, Event-Binding, Conditionals, Loops
- Templates, Components, Mixins
- Transitions, Lifecycle Hooks, Computed

Additional Packages (still part of the official Vue.js project):

- Routing
- Streaming Server Side Rendering with Client Side Hydration
- TypeScript as alternative to JavaScript
- JSX-templates as alternative to HTML-Templates
- Flux/React style applications with vuex

1. https://ssr.vuejs.org/en/
2. https://vuejs.org/v2/guide/typescript.html
3. https://github.com/vuejs/babel-plugin-transform-vue-jsx
4. https://vuex.vuejs.org/en/

Vue.js – developer friendly, fast and versatile

1 Expectations – Tools, Ecosystem, Options

2 **Basics – Getting Started**

3 Projects – Components and Structure

4 Performance – From different Angles

5 Lessons Learned – Looking back

Initialization of Vue.js

1. Define DOM-Area
2. Load Vue.js
3. Initialize Vue.js

```
<div id="app">
    {{ message }}
</div>
<script src="js/vue.js"></script>
<script>
var app = new Vue({
    el: '#app',
    data: {
        message: 'Hello Vue.js!'
    }
})
</script>
```

Minimal Example

- View-, Input- and Event-Bindung
- Conditionals

```
<div id="app">
    <p>{{ message }}</p>
    <p v-if="message === ''">
       Please enter a Message!</p>
    <p>
      <input v-model="message">
    </p>
    <button
       v-on:click="reverseMessage">
       Reverse Message
    </button>
</div>
```

```
var app = new Vue({
    el: '#app',
    data: {
        message: 'Hello Vue.js!'
    },
    methods: {
        reverseMessage: function () {
            this.message =
                this.message
                   .split('')
                   .reverse().join('')
        }
    }
})
```

Vue.js – developer friendly, fast and versatile

1 Expectations – Tools, Ecosystem, Options

2 Basics – Getting Started

3 **Projects – Components and Structure**

4 Performance – From different Angles

5 Lessons Learned – Looking back

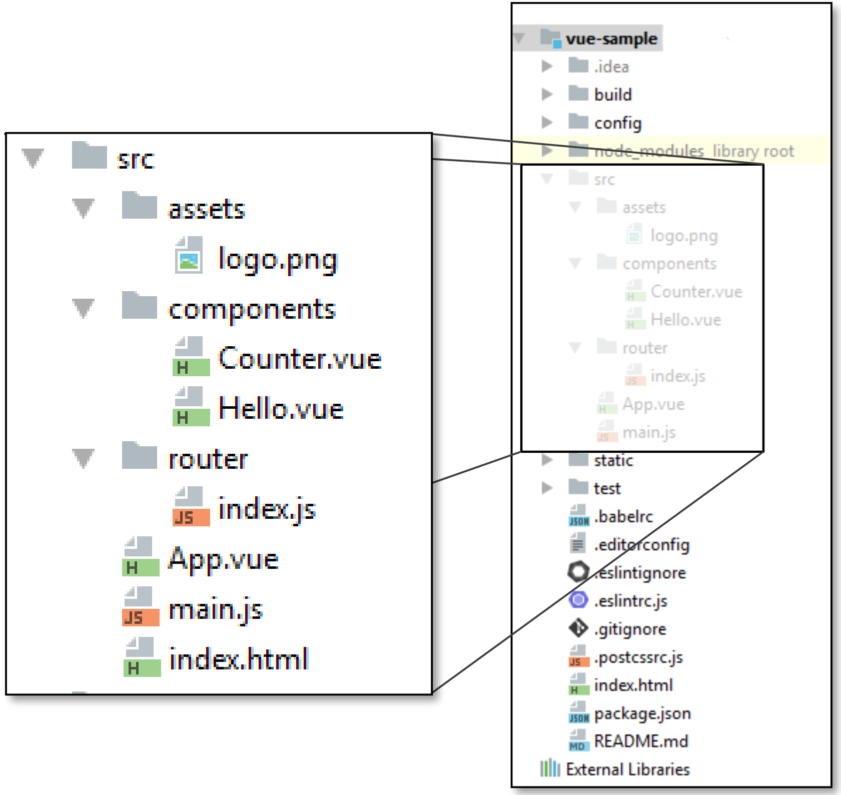
Standard Project Structure

Create structure:

```
npm install -g vue-cli
vue init ...
```

Run for development including hot reload:

```
npm install
npm run dev
```



Single File Components

For every component HTML, JavaScript and CSS in one single file.

This gives you good overview of all resources of a page.

The IDE provides syntax highlighting for all three parts.

Navigation between parts via your IDE.

```
<template>
  <div class="hello">
      <!-- ... ->
  </div>
</template>

<script>
export default {
  /* ... */
}
</script>

<style scoped>
h1, h2 {
  font-weight: normal;
}
</style>
```

Router switches between components

Different routes lead to different components.
When the URL changes, the displayed component changes as well.

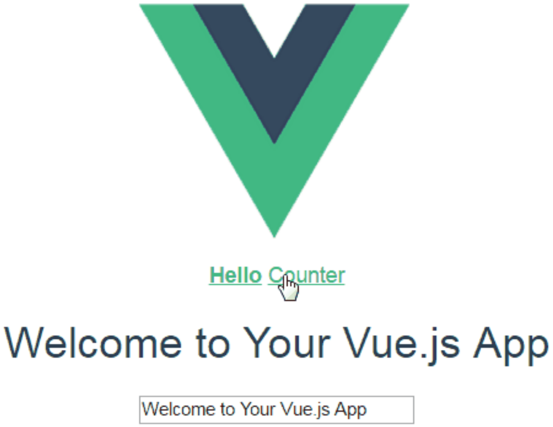
```
Vue.use(Router)

export default new Router({
  routes: [
    {
      path: '/',
      name: 'Hello',
      component: Hello
    },
    {
      path: '/counter',
      name: 'Counter',
      component: Counter
    }
  ]
})
```

```
<template>
  <div id="app">
    <router-view></router-view>
  </div>
</template>
```

Transitions between states and views

Transitions for the data displayed (states) or for the HTML shown (views).



Hello Counter

Welcome to Your Vue.js App

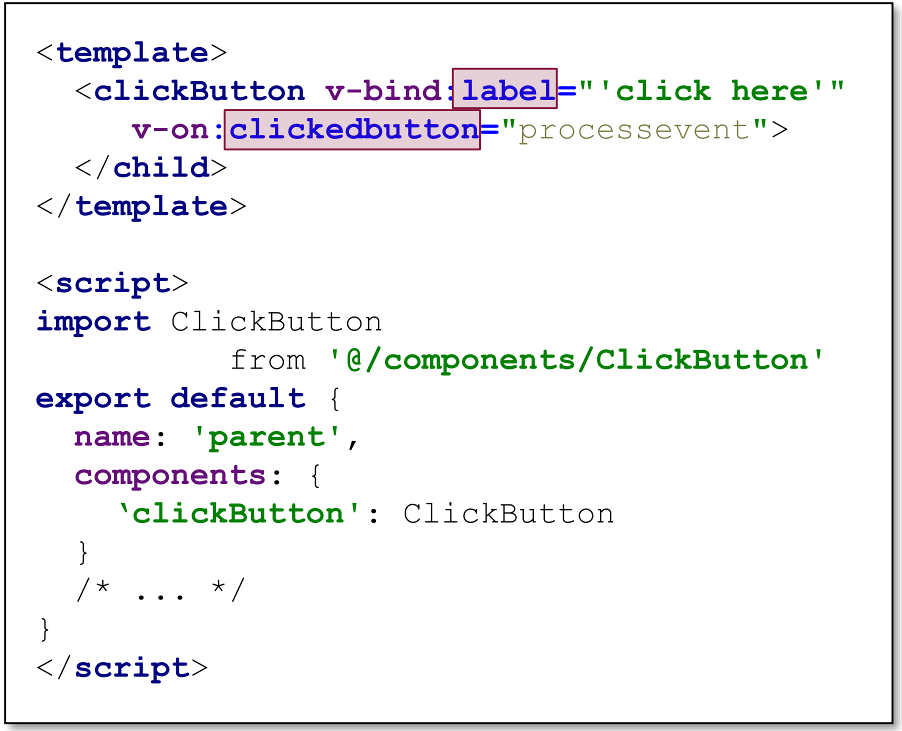
Welcome to Your Vue.js App

```
<template>
  <div id="app">
    <transition name="slide-fade"
                mode="out-in">
      <router-view></router-view>
    </transition>
  </div>
</template>

<style>
.slide-fade-enter-active {
  transition: all .3s ease;
}
/* ... */
</style>
```

1. Sarah Drasner @ Software Engineering Daily https://softwareengineeringdaily.com/2017/12/01/animating-vuejs-with-sarah-drasner/
2. Vue Guide on Transitions: https://vuejs.org/v2/guide/transitions.html

Parent-Child relations for components

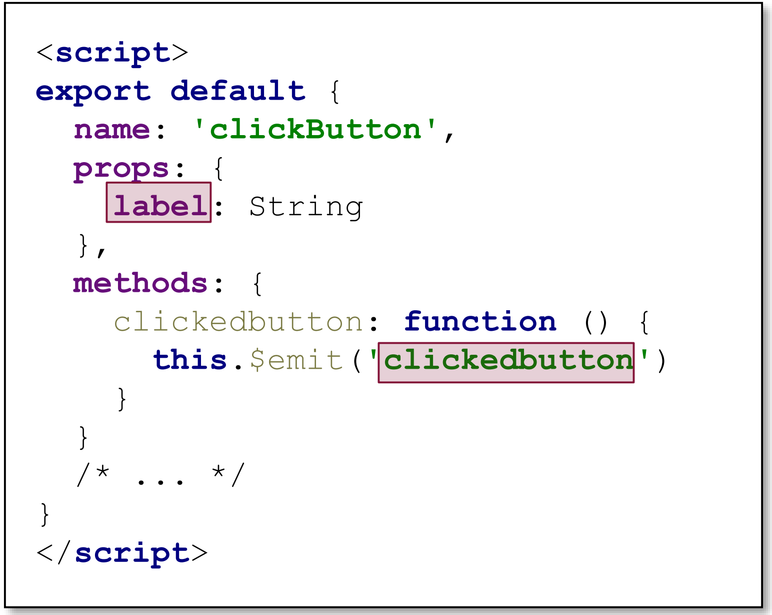


```
<template>
  <clickButton v-bind:label="'click here'"
     v-on:clickedbutton="processevent">
  </child>
</template>

<script>
import ClickButton
          from '@/components/ClickButton'
export default {
  name: 'parent',
  components: {
    ‘clickButton': ClickButton
  }
  /* ... */
}
</script>
```

The parent component passes props to the child.

Parent can listen for custom events of the child.



```
<script>
export default {
  name: 'clickButton',
  props: {
    label: String
  },
  methods: {
    clickedbutton: function () {
      this.$emit('clickedbutton')
    }
  }
  /* ... */
}
</script>
```

Slots to override default sub-templates of the child

The parent passes slots that can override pre-defined parts of the HTML displayed by the child.

The child defines props that can be used in the slots to display data.

```
<!-- parent -->
<definition term="term two"
     description="description 2">
  <template slot="slot_term"
            scope="props">
          T: {{ props.term }}
  </template>
</definition>
```

```
<!-- child with defaults -->
<template>
  <span>
    <dt>
      <slot name="slot_term"
           :term="term">
        {{term}}
      </slot>
    </dt>
    <dd>
      <slot name="slot_description"
           :description="description">
        {{description}}
      </slot>
    </dd>
  </span>
</template>
```

Vue.js – developer friendly, fast and versatile

1 Expectations – Tools, Ecosystem, Options

2 Basics – Getting Started

3 Projects – Components and Structure

4 **Performance – From different Angles**

5 Lessons Learned – Looking back

Performance during Development

Learnability

- Online resources, few simple concepts

Comprehend code fellow developers wrote

- simple project structure and single-file components

Fast round-trip developing

- hot-reload in development mode
- components and state visible with Vue.js Developer Tools in Chrome

Performance in Production

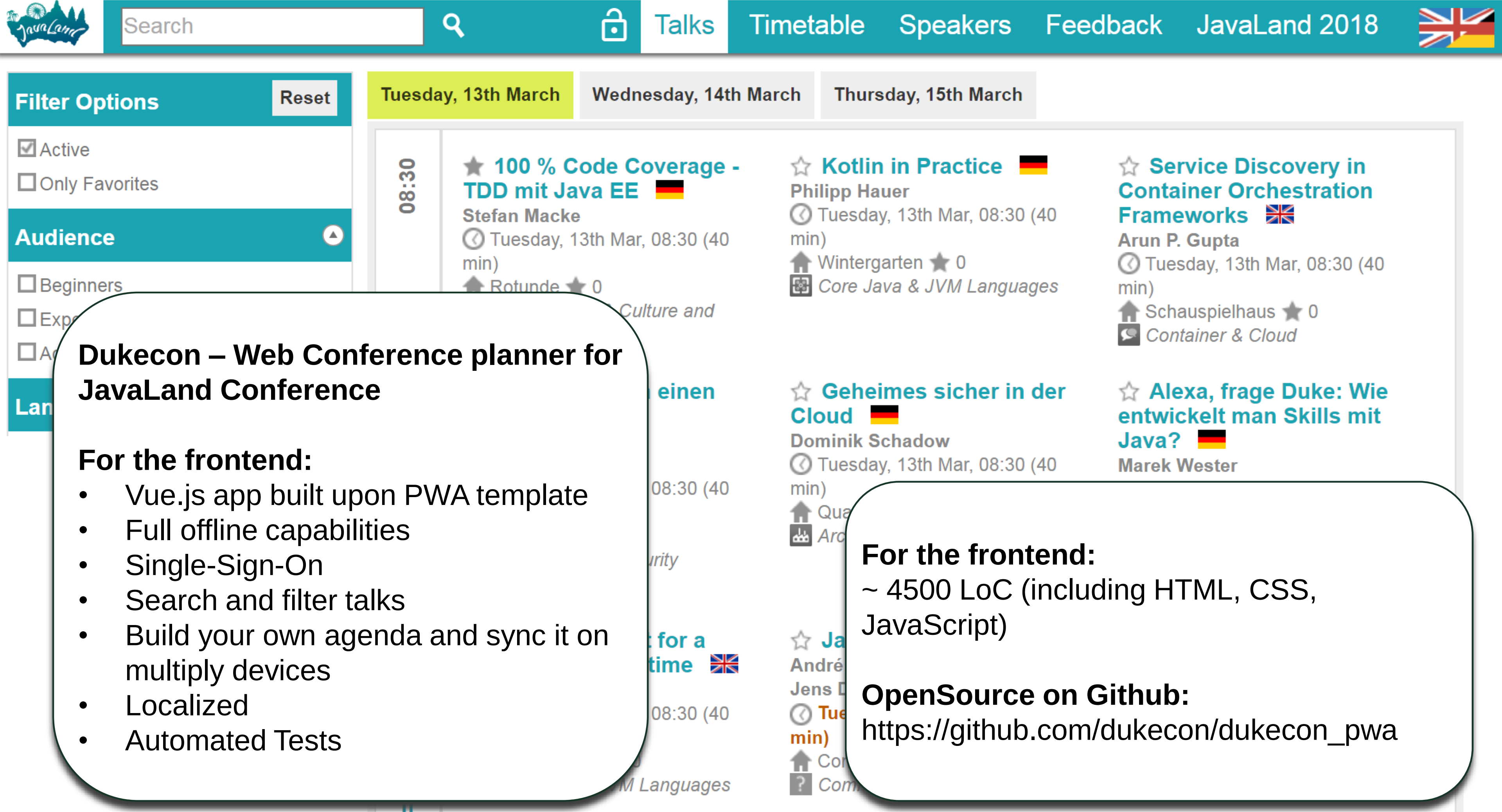
Stumbling blocks

- New properties of the model needs to be placed with *Vue.set* to enable change tracking.

Load-Times and Rendering

- Small size: Vue.js has 20k min+gzip
  (no other dependencies, templates are pre-compiled during build)
- No optimization for rendering necessary, event for big tables
- Great performance benchmarks

1. http://www.stefankrause.net/js-frameworks-benchmark6/webdriver-ts-results/table.html



☑ Active
☐ Only Favorites
☐ Beginners

Tuesday, 13th March | Wednesday, 14th March | Thursday, 15th March

08:30

★ 100 % Code Coverage - TDD mit Java EE
Stefan Macke
Tuesday, 13th Mar, 08:30 (40 min)
Rotunde ★ 0

☆ Kotlin in Practice
Philipp Hauer
Tuesday, 13th Mar, 08:30 (40 min)
Wintergarten ★ 0
Core Java & JVM Languages

☆ Service Discovery in Container Orchestration Frameworks
Arun P. Gupta
Tuesday, 13th Mar, 08:30 (40 min)
Schauspielhaus ★ 0
Container & Cloud

☆ Geheimes sicher in der Cloud
Dominik Schadow
Tuesday, 13th Mar, 08:30 (40 min)

☆ Alexa, frage Duke: Wie entwickelt man Skills mit Java?
Marek Wester

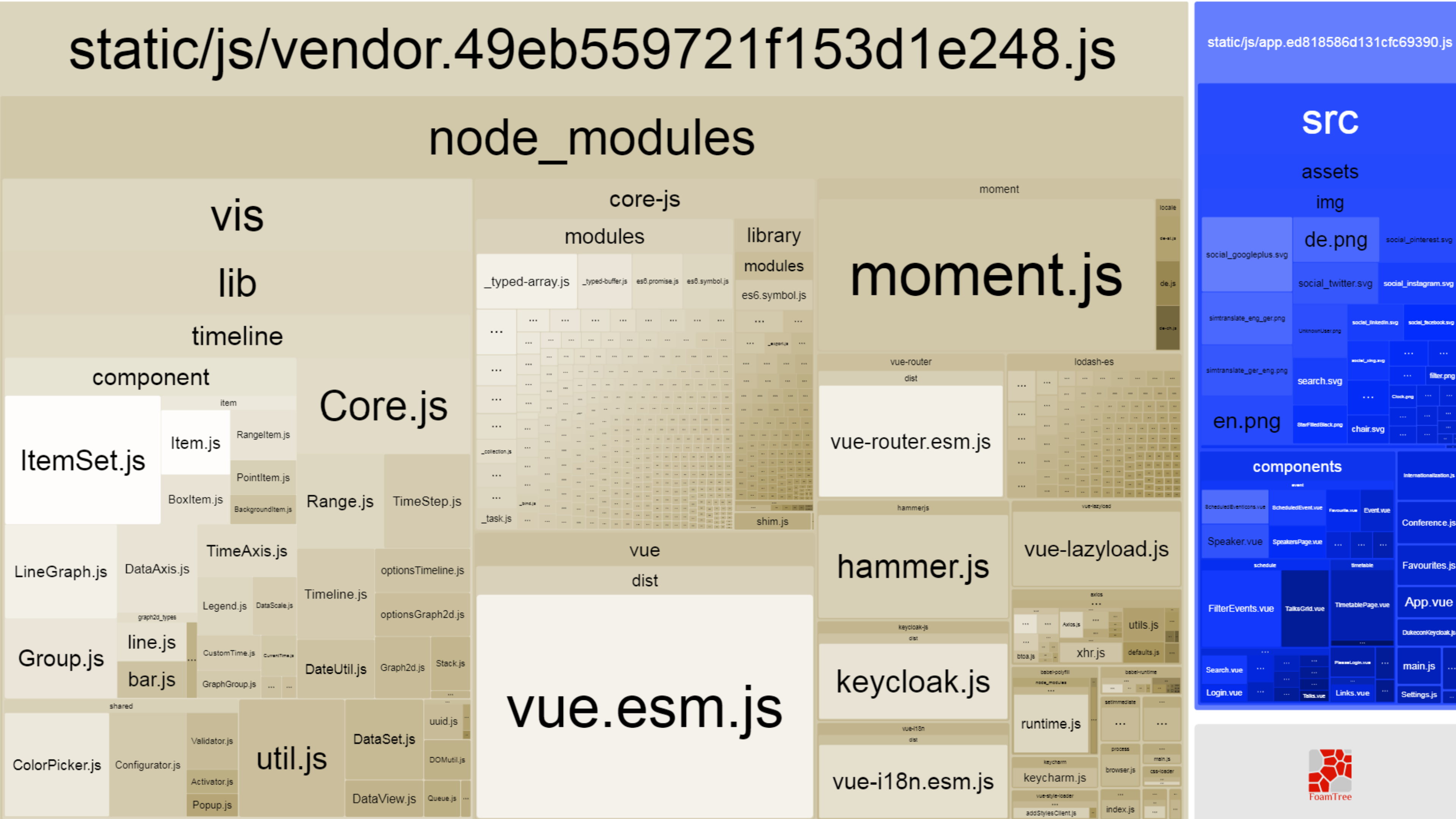
Dukecon – Web Conference planner for JavaLand Conference

For the frontend:

- Vue.js app built upon PWA template
- Full offline capabilities
- Single-Sign-On
- Search and filter talks
- Build your own agenda and sync it on multiply devices
- Localized
- Automated Tests

For the frontend:
~ 4500 LoC (including HTML, CSS, JavaScript)

OpenSource on Github:
https://github.com/dukecon/dukecon_pwa


static/js/vendor.49eb559721f153d1e248.js
node_modules
vis
lib
timeline
component
item
ItemSet.js
Item.js
RangeItem.js
PointItem.js
BoxItem.js
BackgroundItem.js
Core.js
Range.js
TimeStep.js
LineGraph.js
DataAxis.js
TimeAxis.js
graph2d_types
Group.js
line.js
bar.js
Legend.js
DataScale.js
CustomTime.js
CurrentTime.js
GraphGroup.js
Timeline.js
optionsTimeline.js
optionsGraph2d.js
DateUtil.js
Graph2d.js
Stack.js
shared
ColorPicker.js
Configurator.js
Validator.js
Activator.js
Popup.js
util.js
DataSet.js
DataView.js
uuid.js
DOMutil.js
Queue.js
core-js
modules
_typed-array.js
_typed-buffer.js
es6.promise.js
es6.symbol.js
_collection.js
_task.js
_bind.js
library
modules
es6.symbol.js
_export.js
shim.js
vue
dist
vue.esm.js
moment
moment.js
locale
de-at.js
de.js
de-ch.js
vue-router
dist
vue-router.esm.js
lodash-es
hammerjs
hammer.js
vue-lazyload
vue-lazyload.js
keycloak-js
dist
keycloak.js
vue-i18n
dist
vue-i18n.esm.js
axios
Axios.js
utils.js
btoa.js
xhr.js
defaults.js
babel-polyfill
node_modules
babel-runtime
runtime.js
setimmediate
keycharm
keycharm.js
process
browser.js
main.js
css-loader
vue-style-loader
addStylesClient.js
index.js
static/js/app.ed818586d131cfc69390.js
src
assets
img
social_googleplus.svg
de.png
social_pinterest.svg
social_twitter.svg
social_instagram.svg
simtranslate_eng_ger.png
UnknownUser.png
social_linkedin.svg
social_facebook.svg
simtranslate_ger_eng.png
search.svg
social_xing.svg
filter.png
Clock.png
en.png
StarFilledBlack.png
chair.svg
components
event
ScheduledEventIcons.vue
ScheduledEvent.vue
Favourite.vue
Event.vue
Internationalization.js
Conference.js
Speaker.vue
SpeakersPage.vue
Favourites.js
schedule
timetable
FilterEvents.vue
TalksGrid.vue
TimetablePage.vue
App.vue
DukeconKeycloak.js
Search.vue
PleaseLogin.vue
main.js
Login.vue
Talks.vue
Links.vue
Settings.js
FoamTree

Vue.js – developer friendly, fast and versatile

1 Expectations – Tools, Ecosystem, Options

2 Basics – Getting Started

3 Projects – Components and Structure

4 Performance – From different Angles

5 Lessons Learned – Looking back

Expectations met

- Newbies got started quickly.
- Complex GUI delivered with great runtime performance and a maintainable structure.
- Components help structuring the application.
- We developed own shared components for navigation and tables.

Update from Vue.js 1.x to Vue.js 2.x:

- Minimal Vue.js changes, aided by upgrade and guide migration helper script provided by Vue.js (some changes could already be implemented while using 1.x)
- Benefits: faster rendering, better warning messages and error handling especially when developing
- Medium effort for Webpack upgrade including scripts

Links

Vue.js
https://vuejs.org/

Vue-router
https://router.vuejs.org/en/

Vuex
https://vuex.vuejs.org/en/

The Majesty of Vue.js 2
https://leanpub.com/vuejs2

Awesome Vue.js
https://github.com/vuejs/awesome-vue

Vue Dev Tools for Google Chrome
https://github.com/vuejs/vue-devtools

Sarah Drasner @ Software Engineering Daily
https://softwareengineeringdaily.com/2017/12/01/animating-vuejs-with-sarah-drasner/

Dukecon PWA
https://github.com/dukecon/dukecon_pwa

@ahus1de

Alexander Schwartz
Principal IT Consultant

+49 171 5625767
alexander.schwartz@msg.group

@ahus1de

msg systems ag (Headquarters)
Robert-Buerkle-Str. 1, 85737 Ismaning
Germany

www.msg.group

.msg .consulting .solutions .partnership