JavaScript & Vue

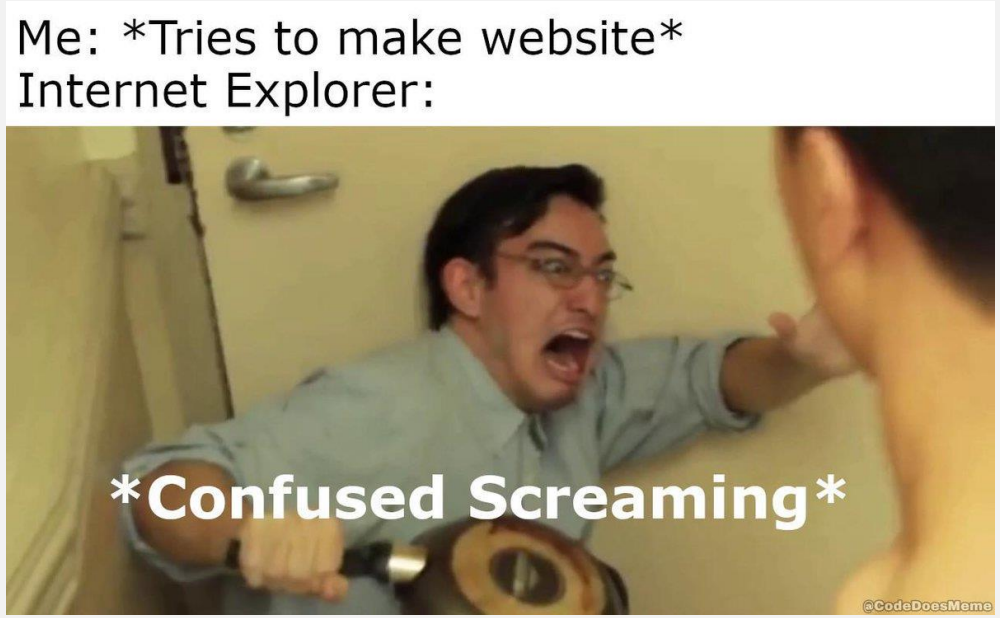
Inhalt

- Theorie
- Installation
- Praktisches
- After Work Vollsuff im HQ

Theorie

Was ist JavaScript (JS)

- Problem: HTML ist keine Programmiersprache
  - 1995 wollte man eine „Art“Java im Browser haben
- Standardisiert unter dem Namen „ECMAScript“ (ES)
  - Aktuelle Version: ECMAScript 2019
  - Problem: Browsercompatibilität
  - Nutzbare Version: ES2015 (ES6)



Wie sieht das aus?

```
<!DOCTYPE html>
<html>

<body>

    <h2>JavaScript in Body</h2>

    <p id="demo">A Paragraph.</p>

    <button type="button" onclick="myFunction()">Try it</button>

    <script>
        function myFunction() {
            document.getElementById("demo").innerHTML = "Paragraph changed.";
        }
    </script>

</body>

</html>
```

JavaScript Eigenschaften

- Keine expliziten Typen !!!!11 (string, int, …)
  - Alles ist nach außen ein Objekt, intern aber nicht
  - Alles kann verglichen werden: Funktionen, Klassen, Variablen
  - JavaScript konvertiert selbst intern so, dass es passt

```
> 20000 + "miles"
"20000miles"
> true + "false"
"truefalse"
> "R" + 2 + "D" + 2
"R2D2"
>
```

undefined==false	false
null==false	false
undefined==0	false
null==0	false
undefined==''	false
null==''	false
false==''	true
''==0	true
false==0	true
null==undefined	true

JavaScript nutzen

- If abfragen immer immer mit === oder !== nutzen
  - Typensicherer, da wenig „automatische“ Konvertierung stattfindet

- Variable deklarieren: var meinText = „hallo“;

JSON

- Datenspeicherformat ähnlich zu XML
- Kann 1zu1 in JS genutzt werden

```
{
  "Gruppenname": "Mitglieder",
  "user": [
    { "id": 1, "name": "Max" },
    { "id": 10, "name": "Mark" },
    { "id": 11, "name": "Manuel" },
    { "id": 11, "name": "Marcel" },
    { "id": 100, "name": "Moriz" }
  ]
}
```

- `var variable = {...};`
- `Inhaber.Alter = 19;`
- `if(user[0].name === „Max“){`
  `...`
  `}`

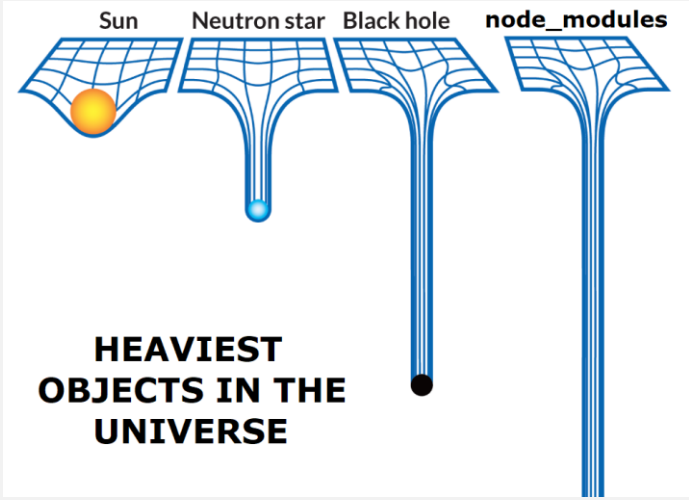
```
{
    "Herausgeber": "Xema",
    "Nummer": "1234-5678-9012-3456",
    "Deckung": 2e+6,
    "Waehrung": "EURO",
    "Inhaber":
    {
      "Name": "Mustermann",
      "Vorname": "Max",
      "maennlich": true,
      "Hobbys": ["Reiten", "Golfen", "Lesen"],
      "Alter": 42,
      "Kinder": [],
      "Partner": null
    }
  }
```

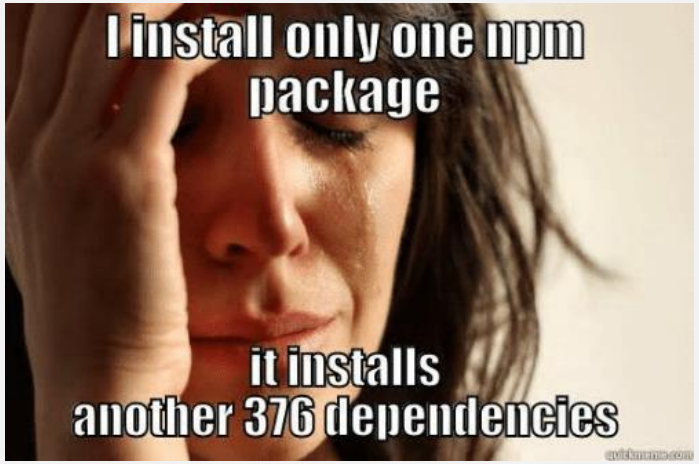
NPM

- Das Maven der JavaScript Welt
  - nur in geil
- Befehl: `npm install <package-name>`
- Erzeugt eine package.json config (die nicht Kacke ist!)

NPM

- Für viele nur eine andere Art Krebs
- Da alle Quelldateien wiederum JavaScript Dateien sein müssen eskaliert die Anzahl oftmals.
- Beispiel OPhIT
  - über 900 Pakete
  - Über 50.000 Dateien





Und was soll Node.js sein?

- JavaScript als Server ohne Browser
- Chrome Engine als Basis im Hintergrund
  - Und damit fast auf jedem System läuffähig
- In Node.js läuft Vue
- Interessiert uns eigentlich nicht

Vue

- Vue ist ein JavaScript „Framework“
  - In JS geschrieben und erzeugt JS, HTML und CSS Dateien
  - Erweitert HTML, sodass wir mehr abstrakten Code schreiben können

  - Beispiel:
    - Irgendwo im Code haben wir die Variable `var user = { „name“: „max“}`

    - Wir können nun ein einen HTML Button bauen der nur angezeigt wird, wenn dieser User existiert:

```
<button v-if=„user.name === ‚max‘“>
  Click Me
</button>
```

Aufbau

HTML

JavaScript

CSS style code

```
<template>
  <div>Ganz normaler Text</div>
</template>

<script>
export default {
  data: {
    user: [
      { id: 1, name: "Max" },
      { id: 10, name: "Mark" },
      { id: 11, name: "Manuel" },
      { id: 11, name: "Marcel" },
      { id: 100, name: "Moriz" }
    ]
  },
  methods: {
    clickFunction() {
      console.log("click happend!");
      return "you clicked me";
    }
  }
};
</script>

<style>
div {
  margin: 8px;
}
</style>
```

Aufbau II

Variablen

Methoden

```
<template>
  <div>Ganz normaler Text</div>
</template>

<script>
export default {
  data: {
    user: [
      { id: 1, name: "Max" },
      { id: 10, name: "Mark" },
      { id: 11, name: "Manuel" },
      { id: 11, name: "Marcel" },
      { id: 100, name: "Moriz" }
    ]
  },
  methods: {
    clickFunction() {
      console.log("click happend!");
      return "you clicked me";
    }
  }
};
</script>

<style>
div {
  margin: 8px;
}
</style>
```

Loop (Array) in Vue

- Temporär wird eine variable erstellt, mit der dann in den Unterobjekten mit `{{tempVariable}}` zugegriffen werden kann
  `v-for=„tempVariable in Variable“`

- Unterscheidung der Objekte
  `:key=„indentifier“`

- Hohzählender Index
  `v-for=„(user, i) in users“`

```
<template>
  <div>
    <div v-for="user in users" :key="user.id">
        Username: {{user.name}}
    </div>
  </div>
</template>

<script>
export default {
  data: {
    users: [
      { id: 1, name: "Max" },
      { id: 10, name: "Mark" },
      { id: 11, name: "Manuel" },
      { id: 11, name: "Marcel" },
      { id: 100, name: "Moriz" }
    ]
  },
  methods: {
    clickFunction() {
      console.log("click happend!");
      return "you clicked me";
    }
  }
};
</script>

<style>
div {
  margin: 8px;
}
</style>
```

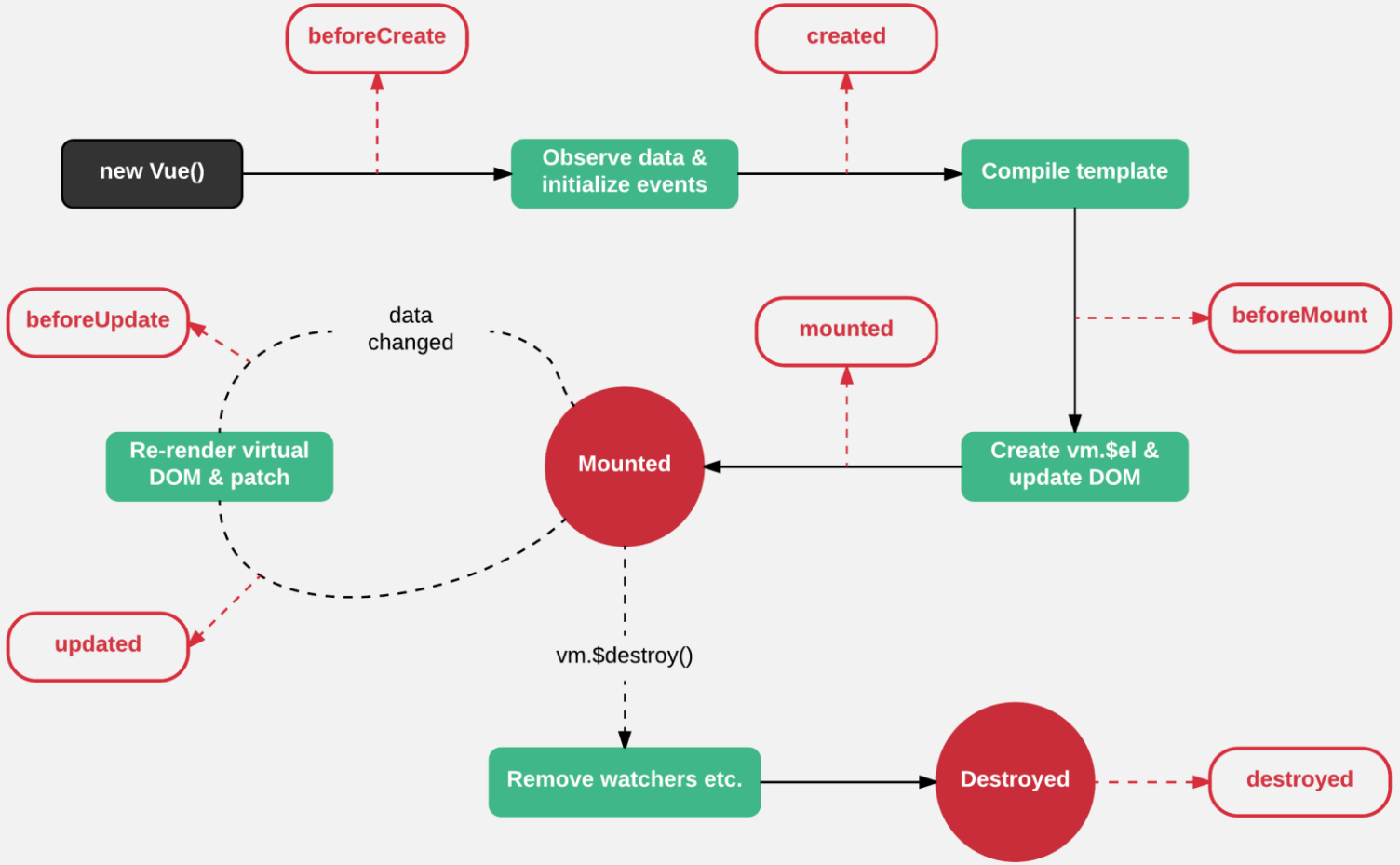
Events

- `@EventName=„FunktionsName“`

```
<template>
  <div>
    <button @click="clickFunction">Click me</a>
  </div>
</template>

<script>
export default {
  methods: {
    clickFunction() {
      console.log("click happend!");
      return "you clicked me";
    }
  }
};
</script>
```

Lifecycle Events

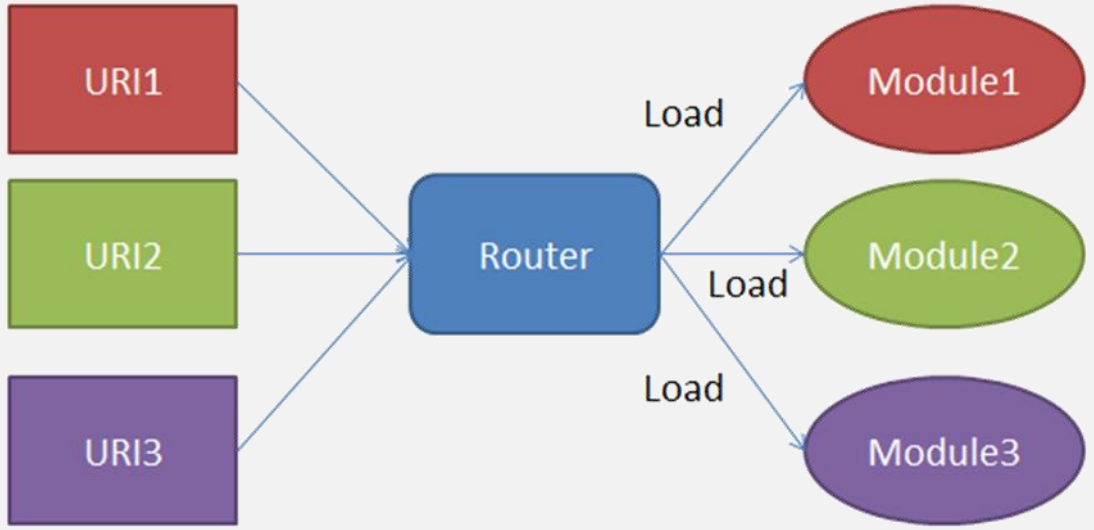


Nur Theorie: Store

- Problem: Wenn die Seite neu geladen wird, sind alle Daten weg
- Lösung: Alles in Cookies speichern
- Ein Store verwaltet diesen Cookie-Speicher für euch
- Viele Unsicherheiten
  - Löschbar
  - Nicht aktiv
  - Datenschutz des EuGH$_{uso}$

Nur Theorie: Router

- Problem: Wir können nicht unser ganzes Programm auf eine Seite klatschen
- Lösung: Wir nutzen einen Router, der für uns die Navigation übernimmt

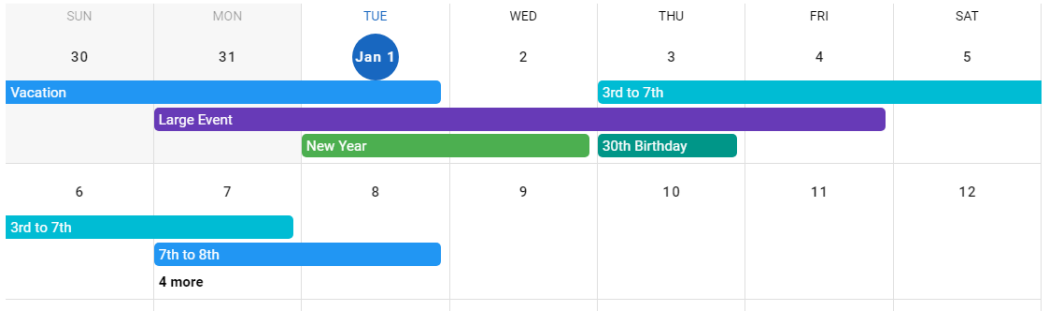


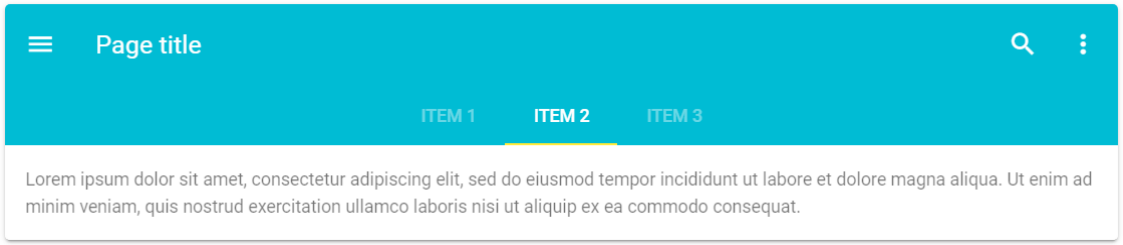
Nur Theorie: Code Splitting

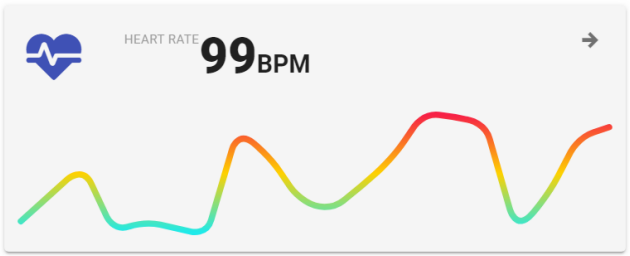
- Problem: Wenn wir große Webseiten schreiben, dann braucht unsere Seite zu lange, um zu laden
- Lösung: Nur den Code laden, wen wir brauchen
  - Den Rest laden wir nach
- Passiert oft schon automatisch
- Führt dazu, dass viele kleine unleserliche Dateien erstellt werden

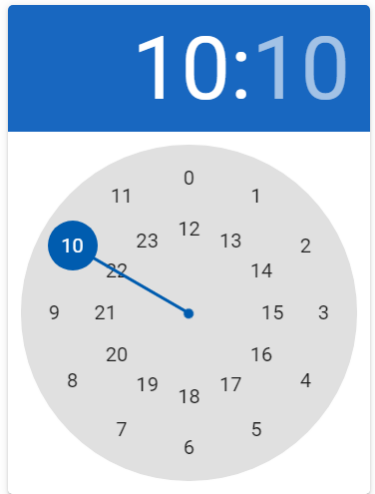
Vuetify

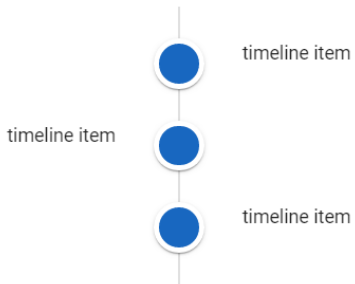
- Noch ein JavaScript Framework
- Baut auf und benötigt Vue
- Dokumentation: https://vuetifyjs.com/en/

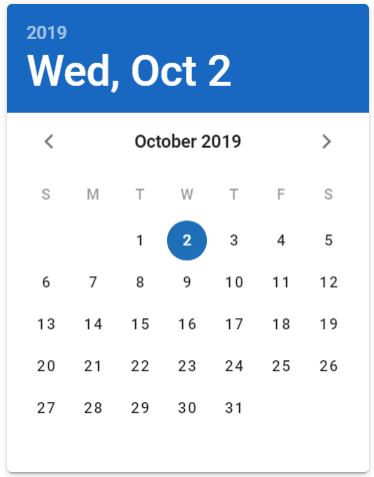


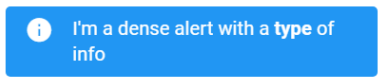


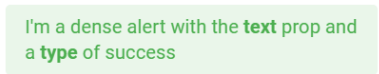


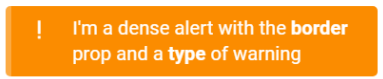


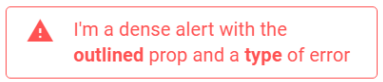


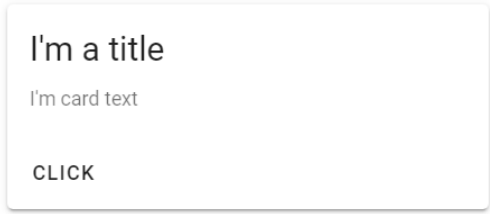


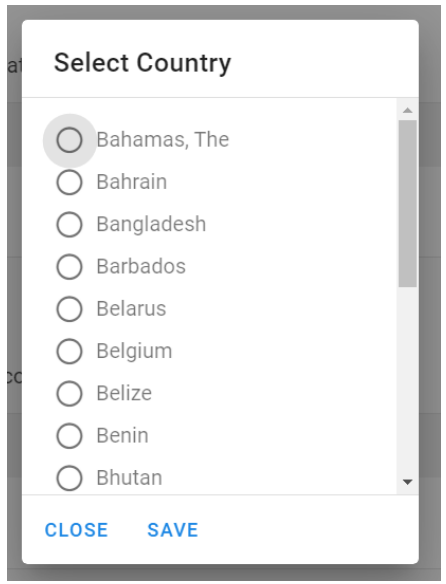


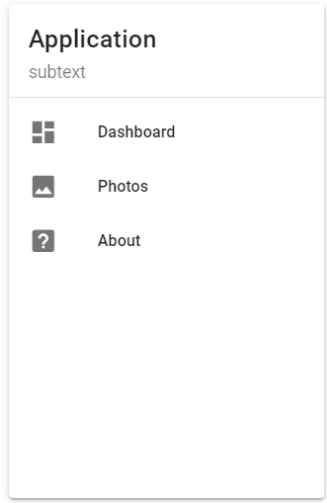








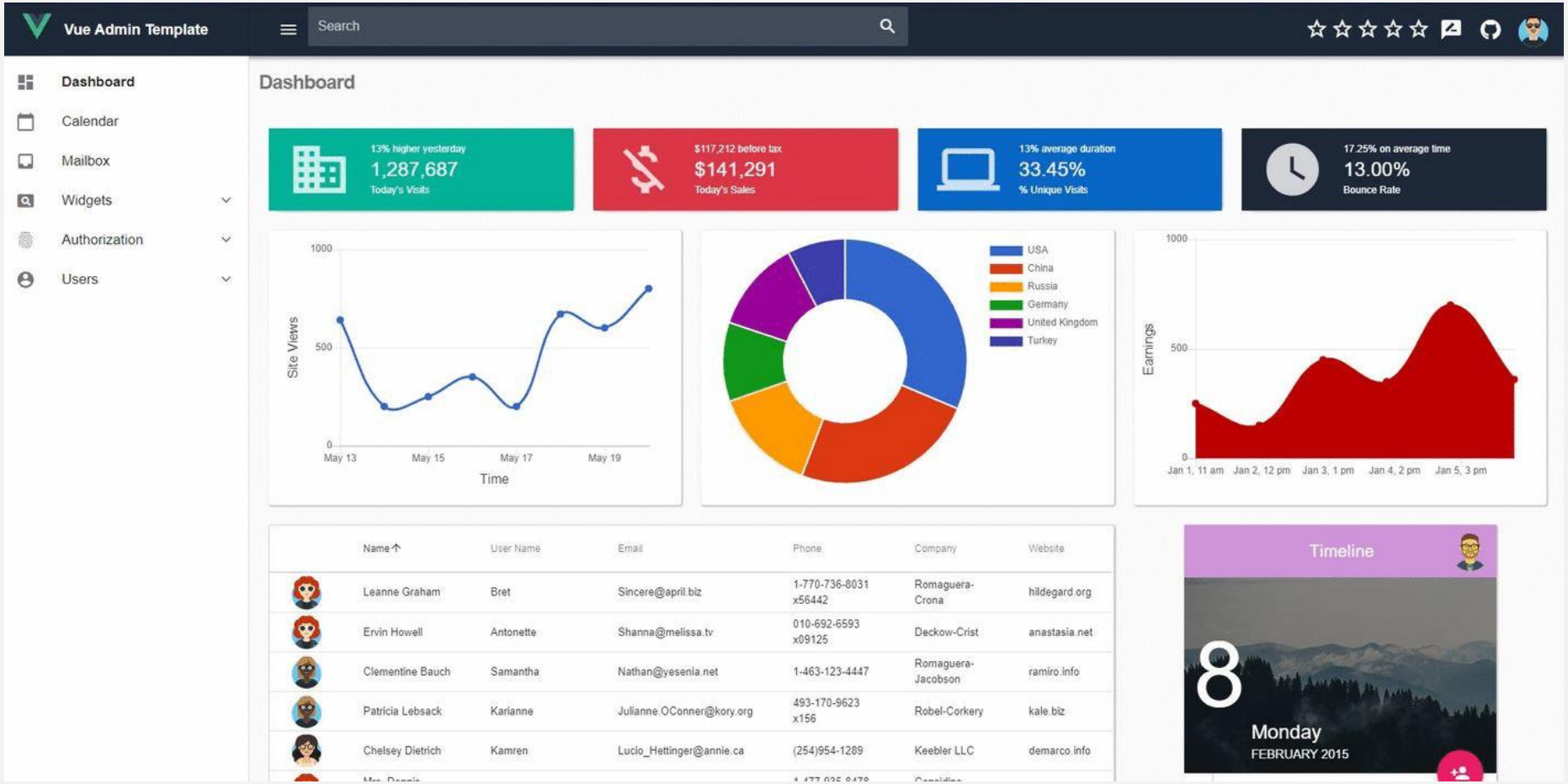




Dessert (100g serving)	Calories	Fat (g)	Carbs (g)	Protein (g)	Iron (%)
Frozen Yogurt	159	6	24	4	1%
Ice cream sandwich	237	9	37	4.3	1%
Eclair	262	16	23	6	7%
Cupcake	305	3.7	67	4.3	8%
Gingerbread	356	16	49	3.9	16%

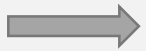
Rows per page: 5 1-5 of 10

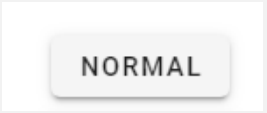
Beispiel


Vue Admin Template
Search
Dashboard
Calendar
Mailbox
Widgets
Authorization
Users
Dashboard
13% higher yesterday
1,287,687
Today's Visits
$117,212 before tax
$141,291
Today's Sales
13% average duration
33.45%
% Unique Visits
17.25% on average time
13.00%
Bounce Rate
Site Views
1000
500
0
May 13
May 15
May 17
May 19
Time
USA
China
Russia
Germany
United Kingdom
Turkey
Earnings
1000
500
0
Jan 1, 11 am
Jan 2, 12 pm
Jan 3, 1 pm
Jan 4, 2 pm
Jan 5, 3 pm
Name
User Name
Email
Phone
Company
Website
Leanne Graham
Bret
Sincere@april.biz
1-770-736-8031 x56442
Romaguera-Crona
hildegard.org
Ervin Howell
Antonette
Shanna@melissa.tv
010-692-6593 x09125
Deckow-Crist
anastasia.net
Clementine Bauch
Samantha
Nathan@yesenia.net
1-463-123-4447
Romaguera-Jacobson
ramiro.info
Patricia Lebsack
Karianne
Julianne.OConner@kory.org
493-170-9623 x156
Robel-Corkery
kale.biz
Chelsey Dietrich
Kamren
Lucio_Hettinger@annie.ca
(254)954-1289
Keebler LLC
demarco.info
Timeline
8
Monday
FEBRUARY 2015

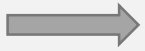
Nutzung von Vuetify

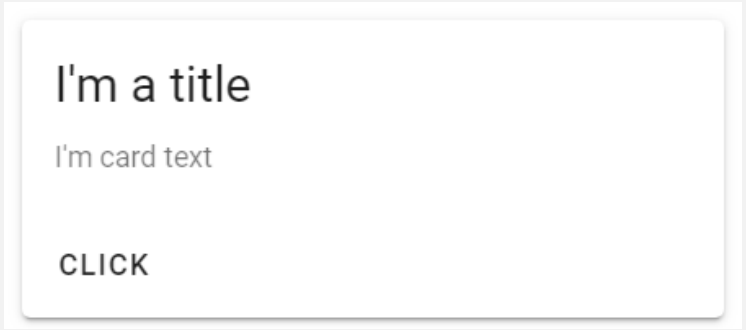
```
<v-btn>Normal</v-btn>
```






```
<v-card>
  <v-card-title>I'm a title</v-card-title>
  <v-card-text>I'm card text</v-card-text>
  <v-card-actions>
    <v-btn text>Click</v-btn>
  </v-card-actions>
</v-card>
```





Installation

Installation

- Das solltet ihr installiert haben:
  - Visual Studio Code (VSC)
  - Node.js

- Ablauf
  - Konsole öffnen (cmd)
  - Befehl zum installieren: `npm install vue`
  - Befehlt zum starten: `vue ui`

  - (Bonus) Plugins *Prettier* und *Vetur* in VSC installieren
  - (Bonus) Chrome Plugin *Vue Devtools* installieren

Installation Teil II

- Projekt anlegen in Vue
  - Tab „Neu“
  - Ordner auswählen
  - Name: Egal
  - Voreinstellung: Standart
- Vuetify installieren
  - Plugins -> plugin hinzufügen -> vue-cli-plugin-vuetify
  - Preset: Default
  - Am Ende die Änderungen mit einer Nachricht akzeptieren (weil GIT)
  - Lädt im Hintergrund eigentlich ein npm package

Installation Teil III

- Projekt starten
  - Aufgabe -> serve -> Aufgabe Starten
  - Warten -> „App öffnen“-Button clicken

Fertig

Code angucken

- Öffnet den Code in VSC und guckt euch die App.vue Datei an.
- Verändert den Code. Die Webseite ändert sich sofort.
  - Das nennt man Hot-Module-Replacement (HMR)
- Öffnet mit einem Rechtsklick die Entwicklereinstellungen in eurem Browser. Guckt mal, was Vue für einen unleserlichen Mist zusammengestellt hat.
  - Viele unnötige und wiederholende Abschnitte
  - Code selbst schreiben ist immer besser
  - Aber eigentlich immer unmöglich

Praktisches Beispiel

Praktisches Beispiel

- Wir bauen ein eine ~~Bierpong App~~ User-Verwaltungs-Seite

Abschließende Worte

Abschließende Worte

- JavaScript ist eine Art „C“ für die Web-Welt
  - Niemand nutzt die reine Low-Level Sprache, weil sie nervt
  - Alle Nutzen aber Tools, die am Ende JavaScript Code produzieren
- Vue ist erfolgreich geworden, weil es nicht so kompliziert wie **Angular** und mächtiger als **React** ist.
  - Angular (Google) und React (Facebook) werden aber viel häufiger genutzt
- Zukünftig werden alle Apps Progessive Web Apps (PWA) sein
  - JavaScript kann alles, was man braucht
  - Auch viele Hardware-Zugriffe (zB. Kamera) sind möglich
  - Performance oft schlecht, aber der Entwicklungsaufwand ist kleiner
  - Apples Safari ist der neue Internet-Explorer, weil er auf viele Standards scheißt
- Einzige Server- und Frontent- Sprache die überall läuft

Dateien

- GIT Repository mit allen Daten: https://github.com/LightSnowDev/JavaScriptVueTutorial-/commits/master

Danke für‘s kommen.

Ab in‘s HQ für n‘ Bier!