

Vue JS

Vue.js ist ein clientseitiges JavaScript Webframework zum Erstellen von Single-Page-Webanwendungen nach dem MVVM-Muster, es kann allerdings auch in Multipage Webseiten für einzelne Abschnitte verwendet werden. Die Entwickler bezeichnen Vue.js als progressiv und inkrementell adaptierbar im Vergleich zu anderen Webframeworks. Dies erlaubt dem nutzenden Entwickler, die Struktur der Anwendung nach eigenen Ansprüchen zu richten.

Geschichte

Evan You war Creative Coder bei Google Creative Labs und entwickelte aus eigenem Interesse Vue Js. Nach der Veröffentlichung 2014 nahm die Beliebtheit der Bibliothek Fahrt auf. Das Release der Version 2 im September 2016 hat Vue.js dann auch unter großen Firmen etabliert.

Funktionen

- **Virtual DOM**

Als einen virtuellen DOM kann man sich ein abstrakte Duplikat des eigentlichen DOMs vorstellen. Die Größe der Kopie ist um einiges geringer, da es nur die Bestandteile enthält, die sich tatsächlich auch verändert können. Sobald DOM-Updates erfolgen sollen, werden diese zuerst einzeln im Virtual DOM vorgenommen. Ein intelligenter Algorithmus vergleicht im Anschluss die Unterschiede in einem Zug im tatsächlich gerendertem DOM.

- **Single File Components**

Vue.js setzt auf Single File Components. Durch dynamisches Bundling ist es möglich, HTML, JavaScript und CSS in getrennte Dateien abzulegen — dennoch gilt hier der Grundsatz einer abgeschlossenen Komponente. In der Regel werden alle drei Bestandteile in eine Datei geschrieben. Kleine Komponenten haben den Vorteil, testbar, wartbar und leicht erweiterbar zu sein. Diese Komponenten folgen daher dem Prinzip des Separation of Concerns. Dieses Prinzip bedeutet, dass man Zuständigkeiten strikt voneinander trennt. Eine einzelne Komponente erfüllt genau eine Aufgabe, nicht mehr und nicht weniger. Dadurch kann gewährleistet werden, dass die Komponente universell einsetzbar ist. Diese Single File Components lassen sich nun baumartig miteinander verbinden. Der übliche Kommunikationsweg zwischen Komponenten ist, dass die Elternkomponente props an ihre Kinder überreicht und per emit über Events benachrichtigt werden kann.

- **Template**

Vue.js lässt einem generell viel Gestaltungsfreiraum. So spielt es aus technischer Sicht keine Rolle, ob man das Template in HTML, JSX oder auch Pug schreibt. Mit den jeweiligen Einstellungen in der Konfigurationsdatei lassen sich die Dialekte transformieren. Die einfachste Variante, um ein Databinding mit dem Script herzustellen, ist die "Mustache" Syntax, die man bereits von Handlebars kennt:

```
<p>{{ greeting }} World!</p>
```

In den Platzhalter wird automatisch von Vue.js der Wert hello des Data-Attributs eingesetzt. Ändert man im Script den Wert von greeting, passt sich auch das Template automatisch an. Dafür muss der Entwickler nicht mehr den Selektor anfragen und den Wert spezifisch ändern.

Um dynamische Werte an das Tag zu binden, benutzt man ein Shorthand mit einem Doppelpunkt als Prefix, zum Beispiel:

- v-bind:href
- v-bind:src
- :href (mit Shorthand)
- :src (mit Shorthand)

- **Script**

Eine Vue Komponente kann global oder lokal registriert werden. Das ist abhängig von der Architektur des Projektes. Generell können dem Script unter anderem folgende Optionen übergeben werden:

- Datengebunden: Hierunter fallen reaktive Datenattribute und Methoden, die mit den Schlüsselwörtern data, props, propsData, computed, methods und watch angewendet werden können.
- DOM-gebunden: Je nachdem, wie man das HTML an die Komponente binden möchte, kann man Optionen für el, template und render verwenden.
- Lifecycle Hooks: Neben den computed values und methods gibt es weitere Methoden, die von Vue.js zu bestimmten Zeitpunkten ausgeführt werden, wie zum Beispiel created, mounted, updated oder destroyed.

- **Deklaratives Rendering**

Hier wird nun die Mustache Syntax und das Binden von Dateiattributen per v-bind vermisch. Im Prinzip funktionieren sie identisch. Ändert sich der Wert von myHomepage oder message, wird das Template automatisch angepasst. Das Ganze funktioniert ebenfalls in die andere Richtung. Bindet man ein Datenmodell an ein Input Feld, so wird der Wert des Modells automatisch angepasst, sobald der Nutzer ihn verändert.

```
<template>
  <div>
    <a v-bind:href="myHomepage">{{ message }}</a>
  </div>
</template>

<script>
export default {
  name: "MyComponent",
  data() {
    return {
      myHomepage: "https://vannsl.io",
      message: "Come to my Homepage"
    };
  }
};
</script>
```

- **Style**

Auch der Style muss nicht in purem CSS3 geschrieben werden. Der Einsatz von SCSS oder SASS ist über das Sprachattribut setzbar. Zusätzlich kann das Schlüsselwort scoped gesetzt werden. Der Vue Loader kann dieses Attribut dazu nutzen, um die dort gesetzten Styles nur auf spezifisch bei der Komponente einzusetzen.

- **Wiederverwendbarkeit**

Um Komponenten dynamisch und wiederverwendbar zu gestalten, gibt es die Möglichkeit, Mixins und Slots einzusetzen. Mixins werden dabei in der Form von einer Vue Komponente geschrieben und tatsächliche Komponenten können diese Mixins integrieren, um den Funktionsumfang ebenfalls zu nutzen. Auf der anderen Seite sind Slots etwas wie Platzhalter. So kann die Elternkomponente einen beliebigen Inhalt in den Slot einer Kinderkomponente einfügen, ohne dass die Kindkomponente wissen muss, was dieser Inhalt darstellt.

- **Aufteilung in mehreren Dateien**

In Vue.js ist man nicht dazu gezwungen, eine Single File Component auch immer in eine einzelne Datei zu schreiben, auch wenn das dem ursprünglichen Namen widerspricht. Es ist möglich, das HTML, CSS und JavaScript in mehrere Dateien abzulegen. Dadurch könnten CSS und JavaScript ausgelagert und in die Datei des Templates importiert werden.

```
<script src="./my-component.js"></script>
```

Erstellung einer “Hello World”-Applikation

Die “Vue CLI” gibt es aktuell in der dritten Version, die einige Verbesserungen zur zweiten Version aufweist. Es bietet jedem Entwickler unter anderem out-of-the-box Support für:

- Vorkonfiguriertes Webpack 4 Setup mit Hot Module Replacement, Code Splitting, Tree shaking und vielem mehr
- ES2017 Transpiler durch Babel 7 und preset-env
- PostCSS und alle großen Präprozessoren
- Generiertes HTML für das Basis Setup

Darüber hinaus gibt es weitere optionale Integrationen wie TypeScript, PWA, Vue Addons wie Router und Store, Linter und Testing.

Schritt 1: Projekt erstellen `$ vue create project-name`

Schritt 2: Entscheiden welche Features man benutzen möchte

Möchte man im nachhinein doch auch einen Router oder Vue hinzufügen, kann man das ganz einfach über `$ vue add [PLUGINNAME]` machen.

Rest Calls

```
axios  
.get('url')  
.then(response => (this.info = response.data.bpi))  
.catch(error => console.log(error))
```

Popular Vue.js sites

- 9gag
- Behance
- Nintendo
- Gitlab
- Font Awesome

Tooling

- Vue CLI
- VuePress

Der nächste große Akteur im Ökosystem von Vue ist VuePress, ein von Vue betriebener statischer Website-Generator. Ursprünglich als Werkzeug zum Schreiben von technischer Dokumentation entwickelt, ist es jetzt ein kleines, kompaktes und leistungsstarkes, kopfloses CMS.

- Vuex

Die Verwaltung des Statuses ist eines der Hauptprobleme, auf die Entwickler bei der Erstellung von Webanwendungen stoßen. Um dies zu lösen, bietet Vue ein Zustandsverwaltungssystem - Vuex. Es dient als zentraler Speicher für alle Komponenten einer Anwendung, wobei der Zustand nur vorhersehbar verändert werden kann.

- Vue.js devtools

Extension for Chrome and Firefox for debugging Vue.js applications

Geschwindigkeit

Wenn es um Leistung geht, ist Vue ein außergewöhnlich schnelles Werkzeug. Mit unglaublich kleinen 18KB nach dem Gzippen wird diese Technologie wahrscheinlich eine lange Lebensdauer vor sich haben.

Vorteile

- Gesamtleistung: Mit seiner überschaubaren Größe und der Fähigkeit, Teile seiner Technologie schrittweise zu übernehmen, deckt die Leistung einen großen Teil dessen ab, was Vue zu einem großartigen Werkzeug macht.
- Benutzerfreundlichkeit: Im Gegensatz zu anderen Frameworks wie Angular ist Vue leicht zu erlernen, was es sowohl für Anfänger als auch für langjährige Profis attraktiv macht.
- Großartige Dokumentation: Das Team hinter Vue hat viel Mühe in die Dokumentation des Tools gesteckt, die für Benutzer unglaublich hilfreich ist.
- Einfache Projektintegration: Dadurch können Sie Vue sofort in Ihrem Projekt einsetzen.

Community

Da es sich um ein neues Framework handelt, braucht Vue noch Zeit, um seine Gemeinschaft auf die Größe derer zu vergrößern, die für React oder Angular stehen. Jedoch kann man sich auf der offiziellen Vue.js viele Antworten auf Fragen holen oder auch auf anderen Seiten wie z.B.: Stackoverflow, Github.

Lernkurve

Vue bietet eine höhere Anpassbarkeit und ist daher leichter zu erlernen als Angular oder React. Darüber hinaus hat Vue eine Überschneidung mit Angular und React in Bezug auf deren Funktionalität wie die Verwendung von Komponenten. Daher ist der Übergang zu Vue von einem der beiden Systeme eine einfache Option. Die Einfachheit und Flexibilität von

Vue ist jedoch ein zweiseitiges Schwert - es erlaubt schlechten Code, was das Debuggen und Testen erschwert.