Vue.js på 15 minuter

Historik

Evan You (f.d. utveckare i AngularJS) skapade Vue.js 2013

Vue 2.6

Vue 3 kommer snart (alpha finns)

Senaste åren en av de tre största ramverken tillsammans med React & Angular

Översikt

Ramverk för webbapplikationer. Kärnan av Vue är inriktad på att kunna visa dynamisk data i HTML-markup och hantera events.

En massa extra libraries finns, både som Vue-teamet tillhandahåller och en massa tredjepartslibbar.

https://vuejs.org/v2/guide/index.html

Exempel “Hello World” som Single file component

```
<template>
    <div>Hej världen!
        <span v-if=”namn.length > 0”>
            säger {{namn}}
        </span>
    </div>
</template>
```

```
<script>
export default {
    data() {
        return { förnamn: ‘’, efternamn: ‘’ };
    },
    computed: {
        namn() {
            const namn = this.förnamn + ‘ ’ + this.efternamn;
            return namn.trim();
        }
    }
};
</script>
```

Single file component

Filnamnskonvention “<Komponentnamn>.vue”
HelloWorld.vue

HTML-syntax:
Innehållet inom delarna <template> <script> <style> processas av verktygskedjan som tolkar och skapar Vue-komponenter från detta.

Interpolation i template inom {{ }}.
Klarar komplexa JS-uttryck men ej modifiering av data.
{{ produkt.namn }} i lager: {{ iLager(produkt) ? ‘ja’ : ‘nej’ }}.

Direktiv

Börjar med “v-”

v-bind:attr=”data” (förkortat: :attr=”data”)
v-on:event=”handler” (förkortat: @event=”handler”)

v-if=”condition”
v-else

Obs: uttrycket inom “” (:attr=“data” ovan) är i ett JS-context så värden är JS-objekt,Number,array etc

Loopar

`v-for` och `:key` används för loopar

```
<ol>
  <li v-for="todo in todos" :key=”todo.id”>
    {{ todo.text }}
  </li>
</ol>
```

Repeterar elementet (<li>...) för varje todo i arrayen.

Input och v-model

För <input> som används v-model för dual-binding vy <-> modell

v-model (:value + @input)

<input type=”text” v-model=”förnamn”>
i princip samma sak som:
<input type=”text” :value=”förnamn” @input=”förnamn = $event.target.value”>

Komponentdefinition

I <script> definieras komponenten som ES6-modul.

importer: import Komp1 from “@/components/Komp1.vue”;
export default { ….. komponentdefinition };

components: Objekt med andra komponenter som denna beror av.
props: Indata from “omvärlden”
data: funktion som skapar initialt tillstånd.
methods: Objekt med metoder som kan anropas från template eller metoder.
computed: Objekt med metoder som “skapar” motsvarande prop som beräknad
mounted: För köra saker när komponenten är redo. (Monterad)

Uttryck “namn” i <template> och “this.namn” i metoder accessar property “namn” som kan finnas i data, props, methods eller computed.

Komponentdefinition, exempel

```
export default {
    components: [Komp1, Komp2],
    props: [‘indata‘]
    data() {
        return { reaktiv: { x:17 },  reaktiv2: ‘’ };
    },
    methods: {
        metoden(y) {
            this.reaktiv.x = y;
            return this.indata.includes(y);
        }
    },
    computed: {
        namn() {
            return this.indata.length ? this.reaktiv2;
        }
    },
    mounted() {
        console.log(‘Monterad, indata: ‘, this.indata);
    },

};
```

Koppla ihop komponenter och props

I **MinKomponent**

<template>:

```
<AnnanKomponent :saker=”sorteradeSaker()” namn=”kalle” />
```

”sorteradeSaker()” är ett uttryck som evalueras i yttre komponenten, dvs MinKomponent. Resultatet binds till den angivna prop kallad ”saker” som definieras i AnnanKomponent.

och i <script>:

```
import AnnanKomponent from '@/components/AnnanKomponent.vue';
export default {
    components: { AnnanKomponent  },
```

Reaktiv data

Komponentens egna tillstånd “data” (Objekt) skapas av angiven funktion (“data” i definitionen) när komponenten skapas.

data-Objektet håller JS-properties som “vaktas” av Vue så att uppdateringar propageras till de som använder datat. S.k reaktiv data.

Detta sker genom att gömda **JS-getters/setters** med Observer-instanser “__ob__” läggs till av Vue i varje reaktivt Objekt. Dvs getters/setters-funktioner anropas under huven för läs/skriv-operationer mot reaktiv data.

På samma sätt är även data i **props** och **computed** props reaktiva.