



Desenvolvimento de Software para WEB

Aula 10 - Introdução ao Vue.js (2)

Professor: Anderson Almada

computed - 1

JS

```
data: {  
  nome: "Jose"  
},  
computed: {  
  maiusculo() {  
    return this.nome.toUpperCase();  
  }  
}
```

HTML

```
Nome: {{nome}} <br>  
Maiusculo: {{maiusculo}}
```

O **computed** também tem o objetivo de manipulação, em que dado a mudança de algum valor de uma chave do objeto JS, ele sempre será chamado.

computed - 2

JS

```
data: {
  primeiroNome: "",
  segundoNome: "",
},
computed: {
  nomeCompleto() {
    return this.primeiroNome + " " +
    this.segundoNome;
  }
}
```

HTML

```
Nome: {{nomeCompleto}} <br>
```

watch

JS

```
data: {
  primeiroNome: ""
},
watch: {
  primeiroNome(novo, velho) {
    console.log("Novo: " + novo);
    console.log("Velho: " + velho);
  }
}
```

HTML

Nome: {{primeiroNome}}

O **watch** é parecido com o **computed**, tendo como diferença que ele escuta uma chave por vez.

components - 1

JS

```
Vue.component('menu-bar', {  
  template: "#menu"  
});  
  
var app = new Vue({  
  ...
```

Os components são
utilizados para o reúso de
de determinados elementos

HTML

```
<template id="menu">  
  <ul>  
    <li>Item 1</li>  
    <li>Item 2</li>  
    <li>Item 3</li>  
  </ul>  
</template>
```

components - 1

HTML

```
<div id="app">  
  <menu-bar></menu-bar>  
  <menu-bar></menu-bar>  
</div>
```

components - 2

JS

```
Vue.component('navbar', {  
  template: "#nav"  
});
```

HTML

```
<template id="nav">  
  <nav class="navbar navbar-expand-sm bg-dark  
  navbar-dark">  
    <a class="navbar-brand" href="#">Logo</a>  
    <menu-bar></menu-bar>  
  </nav>  
</template>
```

Os **components** podem ser interligados.

components - 2

HTML

```
<template id="menu">
  <ul class="navbar-nav">
    <li class="nav-item">
      <a class="nav-link" href="#">Link 1</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Link 2</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Link 3</a>
    </li>
  </ul>
</template>
```


Enviando dados entre components - 1

JS

```
Vue.component('menu-bar', {  
  template: "#menu",  
  props: ['items']  
});
```

Os dados podem ser enviados entre os **components**. Na comunicação, o **props** pode ser utilizado

Enviando dados entre components - 2

HTML

```
<menu-bar v-bind:itens="[ { label: 'Ze'}, { label: 'Chico'}, { label: 'Francisco'} ]"></menu-bar>
```

```
<template id="menu">
  <ul class="navbar-nav">
    <li class="nav-item" v-for="item in itens">
      <a class="nav-link" href="#">{{item.label}}</a>
    </li>
  </ul>
</template>
```

Link a propriedade
com o **v-bind**

Enviando dados entre components - 3

JS

```
Vue.component('navbar', {  
  template: "#nav",  
  data() {  
    return {  
      ok: [{ label: 'Home' }, { label: 'Cadastro' }, { label: 'Contato' }]  
    }  
  }  
});
```

Para enviar dados do pai-filho, utilize os dados, por exemplo

Enviando dados entre components - 4

HTML

```
<template id="nav">
  <nav class="navbar navbar-expand-sm bg-dark navbar-dark" >
    <a class="navbar-brand" href="#">Logo</a>
    <menu-bar v-bind:itens="ok"></menu-bar>
  </nav>
</template>
```

Link a propriedade
com o **v-bind**

Atributos especiais: key

HTML

```
<ul>
  <li v-for="usuario in usuarios"
    :key="usuario.nome">{{usuario.nome}} </li>
</ul>
```

JS

```
data: {
  usuarios: [
    { nome: "Ze" },
    { nome: "Chico" },
    { nome: "Francisco" },
  ]
}
```

Key serve para ajudar o Vue.js a rastrear a identidade de cada nó. Identificador do elemento == id

Atributos especiais: ref - 1

HTML

```
<ul ref="users">
  <li v-for="usuario in usuarios"
:key="usuario.nome">{{usuario.nome}} </li>
</ul>
```

JS

```
data: {
  usuarios: [
    { nome: "Ze" },
    { nome: "Chico" },
    { nome: "Francisco" },
  ]
}
```

Ref serve como referência a um determinado elemento HTML

Atributos especiais: ref - 2

HTML

```
this.$refs.users
```

CONSOLE

```
app.$refs.users
```

Atributos especiais: slot - 1

HTML

```
<template id="nav">
  <nav class="navbar navbar-expand-sm bg-dark navbar-dark" >
    <a class="navbar-brand" href="#">Logo</a>
    <menu-bar v-bind:itens="ok"></menu-bar>
  </nav>
</template>
```

Slot é utilizado nos components para mostrar elementos externos a eles

Atributos especiais: slot - 2

HTML

```
<template id="nav">
  <nav class="navbar navbar-expand-sm bg-dark navbar-dark" >
    <slot></slot>
    <menu-bar v-bind:itens="ok"></menu-bar>
  </nav>
</template>
```

Atributos especiais: slot - 3

HTML

```
<navbar>  
  <a class="navbar-brand" href="#">Logo</a>  
</navbar>
```

JS

```
Vue.component( 'navbar', {  
  template: "#nav",  
  data() {  
    return {  
      ok: [{ label: 'Home' }, { label: 'Cadastro' }, { label: 'Contato' }]  
    }  
  }  
});
```

Atributos especiais: is - 1

HTML

```
<template id="menu-ol">
  <ol class="navbar-nav">
    <li class="nav-item" v-for="item in itens">
      <a class="nav-link" href="#">{{item.label}}</a>
    </li>
  </ol>
</template>

<template id="menu-ul">
  <ul class="navbar-nav">
    <li class="nav-item" v-for="item in itens">
      <a class="nav-link" href="#">{{item.label}}</a>
    </li>
  </ul>
</template>
```

is é utilizado para definir o comportamento de um determinado components anônimos.

Atributos especiais: is - 2

JS

```
Vue.component('menu-bar-ul', {  
  template: "#menu-ul",  
  props: ['itens'],  
});
```

```
Vue.component('menu-bar-ol', {  
  template: "#menu-ol",  
  props: ['itens']  
});
```

Atributos especiais: is - 3

HTML

```
<component :is="template" :items="[ { label: 'UFC' }, { label: 'CRATEUS' }, {  
label: 'RU' } ]" ></component>
```

JS

```
data: {  
  template: "menu-bar-ol"  
}
```

Single Page Application

- Codificar menos do lado server e mais no lado cliente
- Não tem reload de páginas (não precisa baixar)
- Performance
- **SEO**

Vue Router

Instalação

```
<script src="js/vue.js"></script>
```

```
<script src="https://unpkg.com/vue-router@3.0.2/dist/vue-router.js" ></script>
```

Vue Router

Instalação

```
<script src="js/vue.js"></script>
```

```
<script src="js/vue-router.js"></script>
```


Vue Router - 1

HTML

```
<template id="noticias">
  <div>
    Noticias
  </div>
</template>

<template id="menu-ol">
  <ol class="navbar-nav">
    <li class="nav-item" v-for="item in itens">
      <a class="nav-link" href="#">{{item.label}}</a>
    </li>
  </ol>
</template>
```

Vue Router - 2

JS

```
var primeiro = Vue.component( 'component-noticias', {  
  template: "#noticias",  
});
```

```
var segundo = Vue.component( 'menu-bar-ol', {  
  template: "#menu-ol",  
  props: ['itens']  
});
```

Vue Router - 3

JS

```
var router = new VueRouter({
  routes: [
    { path: "/primeiro", component: primeiro },
    { path: "/segundo", component: segundo }
  ]
});

var app = new Vue({
  el: "#app",
  router
});
```

No Router, coloque os paths e os componentes que ele deve renderizar

Vue Router - 4

HTML

```
<div id="app">  
  <router-link to="/primeiro">Primeiro</router-link>  
  <router-link to="/segundo">Segundo</router-link>  
  <router-view></router-view>  
</div>
```

Os **router-links** é utilizado para fazer o link e o **router-view** será utilizado para renderizar o conteúdo do componente

Vue Router - Mode History

JS

```
var router = new VueRouter({  
  mode: 'history',  
  routes: [  
    { path: "/primeiro", component: primeiro },  
    { path: "/segundo", component: segundo }  
  ]  
});
```

Utilizado para retirar o # das URLs. Deve haver um tratamento do lado servidor para acesso direto ao link pelo navegador

Vue Router - Rota nomeada - 1

JS

```
var router = new VueRouter({  
  mode: 'history',  
  routes: [  
    { path: "/primeiro", name: "primeiro", component: primeiro },  
    { path: "/segundo", name: "segundo", component: segundo }  
  ]  
});
```

Utiliza o **name** para
nomear as rotas

Vue Router - Rota nomeada - 2

HTML

```
<div id="app">  
  <router-link :to="{ name: 'primeiro' }">Primeiro</router-link>  
  <router-link :to="{ name: 'segundo' }">Segundo</router-link>  
  <router-view></router-view>  
</div>
```

Utiliza o nome da rota
e o v-bind para o link
entre os dois

Vue Router - Parâmetros - 1

JS

```
var router = new VueRouter({  
  mode: 'history',  
  routes: [  
    { path: "/primeiro", name: "primeiro", component: primeiro },  
    { path: "/segundo", name: "segundo", component: segundo, props: true }  
  ]  
});
```

Caso o componente precise de parâmetros, use o **props: true** para indicar isso.

Vue Router - Parâmetros - 2

HTML

```
<div id="app">
  <router-link :to="{ name: 'primeiro'}">Primeiro</router-link>
  <router-link :to="{ name: 'segundo', params: {itens: [{ label: 'UFC' }, {
label: 'CRATEUS' }, { label: 'RU' }]]}">Segundo</router-link>
  <router-view></router-view>
</div>
```

Vue Router - Redirecionamento - 1

JS

```
// Redirecionar para uma página default  
router.replace('/primeiro');
```

Utilizado para
redirecionar para uma
rota logo no início

Vue Router - Redirecionamento - 2

JS

```
var app = new Vue({  
  el: "#app",  
  router,  
  methods: {  
    clique() {  
      this.$router.push({ name: 'segundo', params: {itens: [{ label: 'UFC' }, {  
label: 'CRATEUS' }, { label: 'RU' } ]}}});  
    }  
  }  
});
```

Utilizado para
redirecionar baseado
em um click, por
exemplo

Estrutura de Diretórios - 1

INSTALAÇÃO

<https://unpkg.com/http-vue-loader@1.4.0/src/httpVueLoader.js>

```
<script src='js/vue.js'></script>  
<script src="js/httpVueLoader.js"></script>
```

Utilizado para organização entre os componentes em diferentes arquivos, para uma melhor manutenção

Estrutura de Diretórios - 2

VUE

```
<template>  
</template>
```

```
<script>  
</script>
```

```
<style>  
</style>
```

Estrutura de Diretórios - 3

OK.VUE

```
<template>
  <div class="hello">
    OK
  </div>
</template>
```

```
<script>
module.exports = {
  name: "ok",
};
</script>
```

OK.VUE

```
<style>
.hello {
  background-color: #ff0;
}
</style>
```

Estrutura de Diretórios - 4

INDEX.JS

```
var Ok = httpVueLoader( './vue/ok.vue' )
```

```
var app = new Vue({  
  el: '#app',  
  components: {  
    ok: Ok  
  },  
  data: {  
    title: "Aula WEB"  
  }  
});
```

Estrutura de Diretórios - 5

HTML

```
<div id="app">
```

```
  {{ title }}
```

```
  <ok></ok>
```

```
</div>
```

```
<script src="js/index.js"></script>
```


Links importantes

- <https://br.vuejs.org/>
- <https://www.youtube.com/watch?v=07-TvnH7XNo&list=PLcoYAcR89n-qq1vGRbaUiV6Q9puy0qigW>



Dúvidas??

E-mail: almada@crateus.ufc.br