



Transformando o menu em componente

Transcrição

Não precisamos meditar muito para ver que há algumas regrinhas para a criação do nosso menu, por isso ele é um forte candidato a ser tornar um componente em nossa aplicação. É exatamente isso que faremos.

Vamos criar o arquivo `alurapic/src/components/shared/menu/Menu.vue`. Ele receberá de um componente pai uma lista de rotas apenas. Vamos aproveitar e colocar uma validação dessa propriedade deixando explicito que aceitamos apenas o tipo `Array`:

```
<!-- alurapic/src/components/shared/menu/Menu.vue -->

<template>
  <nav>
    <ul>
      <li v-for="rota in rotas">
        <router-link :to="rota.path ? rota.path : '/'">
          {{rota.titulo}}</router-link>
        </li>
      </ul>
    </nav>
  </template>
  <script>
    export default {
```

```
    props: {  
      rotas: {  
        type: Array,  
        required: true  
      }  
    }  
  }  
}  
</script>  
  
<style scoped>  
</style>
```

[COPIAR CÓDIGO](#)

Eu preferi chamar internamente de `rota` ao invés de `routes` para não confundir com o array de rotas que se chama `routes`. Mas nada impede que vocês utilizem o mesmo nome.

Agora que temos nosso componente criado, vamos importá-lo em `App`, inclusive precisamos adicioná-lo na propriedade `components` para que seja acessível no template de `App`. Usaremos como nome `meu-menu`:

```
<!-- alurapic/src/App.vue -->  
  
<template>  
  <div class="corpo">  
  
    <meu-menu :rotas="routes"/>  
  
    <transition name="pagina">  
      <router-view></router-view>  
    </transition>  
  
  </div>  
</template>
```

```
<script>
```

```
import { routes } from './routes';  
import Menu from './components/shared/menu/Menu.vue';
```

```
export default {
```

```
  components: {  
    'meu-menu' : Menu  
  },
```

```
  data() {
```

```
    return {
```

```
      routes  
    }
```

```
  }
```

```
}
```

```
</script>
```

```
<style>
```

```
.corpo {  
  font-family: Helvetica, sans-serif;  
  margin: 0 auto;  
  width: 96%;  
}
```

```
.pagina-enter-active, .pagina-leave-active {  
  transition: opacity .3s  
}
```

```
.pagina-enter, .pagina-leave-active {  
  opacity: 0
```

```
}  
</style>
```

[COPIAR CÓDIGO](#)

Excelente, quando a página é recarregada tudo continua funcionando como antes. Agora que temos o menu isolado em um componente, fica mais fácil aplicar aquele estilo profissional ao menu. Mas isso eu deixarei para os meus alunos mestres em CSS.