07

O menu da aplicação

Transcrição

Vamos alterar `alurapic/src/App.vue` e adicionar um menu. Não vamos nos preocupar com estilização do menu, apenas um menu funcional para não perdemos o foco nesta parte tão importante:

```
<!-- alurapic/src/App.vue -->

<template>
  <div class="corpo">

    <nav>
      <ul>
          <li><a href="/">Home</a></li>
          <li><a href="/cadastro">Cadastro</a></li>
      </ul>
    </nav>

    <router-view></router-view>

  </div>
</template>
<script>
</script>
<style>

  /* código omitido */
```

```
</style>
```

COPIAR CÓDIGO

Nosso menu é exibido e quando clicamos no link `Cadastro` somos direcionados para o componente `Cadastro`. Mas nem tudo esta perfeito. Veja que nossas navegações estão disparando o carregamento da página. Se estamos em uma Single Page Application, isso não deveria acontecer, pois já temos tudo o que precisamos carregado. O problema esta no uso da tag `a` para realizar a navegação. Para resolver esse problema, precisamos usar o componente `router-link`:

```
<!-- alurapic/src/App.vue -->

<template>
  <div class="corpo">

    <nav>
      <ul>
          <li><router-link to="/">Home</router-link></li>
          <li><router-link to="/cadastro">Cadastro</router-
link></li>
      </ul>
    </nav>

    <router-view></router-view>

  </div>
</template>
<script>
</script>
<style>
```

```
  /* código omitido */

</style>
```

COPIAR CÓDIGO

Agora sim! Veja que a página não recarregando enquanto clicamos nos itens do menu. Muito melhor! No entanto, se olharmos o arquivo `alurapic/src/routes.js` já temos a lista com todas as rotas definidas nesse arquivo. Lá temos o `path` de cada rota, mas não temos o título. Não tem problema, vamos adicionar a propriedade `titulo` no array de routes. Isso não causará nenhum erro devido a natureza dinâmica do JavaScript e nos permitirá importar esse array para criar nosso menu dinamicamente. Toda vez que uma nova rota for adicionada em routes, automaticamente ela aparecerá como item do menu:

```
// alurapic/src/routes.js

import Home from './components/home/Home.vue';
import Cadastro from './components/cadastro/Cadastro.vue';

// adicionando a propriedade título

export const routes = [

    { path: '', component: Home, titulo: 'Home' },
    { path: '/cadastro', component: Cadastro, titulo:
'Cadastro' }

];
```

COPIAR CÓDIGO

Agora, importando `routes` em `App` e disponibilizando a lista de rotas através da função `data`. Ah, desta vez vamos usar **in** no lugar de **for** na diretiva `v-for`. Eu

prefiro **of**, mas se você vem do Angular 1 pode preferir o **in**:

```
<!-- alurapic/src/App.vue -->

<template>
  <div class="corpo">

    <nav>
      <ul>
          <li v-for="route in routes">
            <router-link :to="route.path ? route.path : '/'">
{{route.titulo}}</router-link>
          </li>
      </ul>
    </nav>

     <router-view></router-view>

  </div>
</template>
<script>

import { routes }  from './routes';

export default {

  data() {

    return {

      routes
    }

  }
```

```
}
</script>
<style>

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

COPIAR CÓDIGO

Vocês devem estar estranhando a linha `<router-link :to="route.path ? route.path : '/'">{{route.titulo}}</router-link>`. Precisamos testar essa condição, porque o `path` de `Home` é um string vazia, mas quando usamos no componente `router-link` precisamos usar um `/`. Aliás, nosso menu é um forte candidato para se tornar um componente, mas não faremos isso agora.