09

Rotas nomeadas

Transcrição

Que tal refletirmos um pouco sobre as rotas da nossa Single Page Application?

Em nosso componente `Cadastro`, temos o botão `VOLTAR` que ao ser clicado dispara uma rota para o endereço `/`. Aprendemos que ele equivale ao `path: ''` de uma de nossas rotas registradas em `alurapic/src/routes.js`.

Mas se a rota para o componente `Home` mudar? Teremos que lembrar de alterá-la o botão `VOLTAR` e em todos os outros lugares da aplicação que estão amarradas ao endereço.

A boa notícia é que podemos dar um nome para as rotas e referenciarmos este nome em todos os lugares que precisamos gerar uma navegação. A vantagem disso é que se o `path` da rota mudar, o restante da nossa aplicação não será afetado.

Vamos alterar o arquivo `routes.js` e adicionar a propriedade `name` para das duas rotas que temos:

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

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

export const routes = [

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

];
```

Agora, no componente `Cadastro`, vamos alterar nosso botão `VOLTAR`. O primeiro passo é que a propriedade `to` da diretiva `router-link` precisa fazer uma data binding com o router global, local de onde ele procurará as rotas. Sendo assim, precisamos adicionar os dois pontos antes da propriedade. Por fim, seu valor é um objeto JavaScript nome temos a propriedade `name` e como valor o nome de uma rota válida:

```
<!-- alurapic/src/components/cadastro/Cadastro.vue -->

<!-- código anterior omitido -->

<router-link :to="{name: 'home'}"><meu-botao rotulo="VOLTAR" tipo="button"/></a>

<!-- código posterior omitido -->
```

Excelente, tudo continua funcionando. Ah, não precisamos alterar o código do nosso menu que é gerado automaticamente, porque se alterarmos algo em nossas rotas, ele refletirá sempre a última versão.

Por fim, o que acontece se acessarmos um endereço que não existe? Apensar o menu da aplicação é exibida. Que tal gerarmos uma navegação para `Home`? Para isso, vamos colocar uma nova rota em `routes.js`:

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

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

export const routes = [

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

];
```

O curinga `*` atribuído à propriedade `path` da nova rota a ativará caso a o endereço acessado não veja o valor de algumas das nossas rotas.

Façamos um teste acessando um endereço que não existe. Somos direcionados para `Home`. No entanto, se você esta tão atento quanto eu, verá que apareceu mais uma opção em nosso menu, mas sem nome. O problema é que a nova rota esta sendo considerada item de menu, o que não deveria acontecer.

Podemos resolver isso adicionando mais uma propriedade na configuração de nossas rotas para indicar se ela deve entrar no menu ou não:

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

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

export const routes = [

    { path: '', name: 'home', component: Home, titulo: 'Home', menu: true },
    { path: '/cadastro', name:'cadastro', component: Cadastro, titulo: 'Cadastro', menu: true },
    { path: '*', component: Home, menu: false }

];
```

Apenas as rotas que tivem `menu: true` serão exibidas em nosso menu. Por fim, precisamos filtrar a lista de rotas para que apenas as que representem menu sejam consideradas:

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

<template>
  <!-- código omitido -->
</template>
<script>

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

export default {

  components: {
```

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

  data() {

    return {

      routes: routes.filter(route => route.menu)
    }

  }

}
</script>
<style>

/* código omitido */
</style>
```

Perfeito!