

Extraindo parâmetro de rota e buscando com nossa API

Transcrição

Como o VueRouter é registrado no global view object temos acesso ao `$route` em nossos componentes. É através dele que conseguimos extrair o parâmetro passado na rota para que possamos utilizá-lo em nosso componente.

O primeiro passo que faremos em `Cadastro` é adicionar a propriedade `id` como propriedade do objeto retornado pela função `data`. Seu valor será `this.$route.params.id`:

```
// alurapic/src/components/cadastro/Cadastro.vue
// código anterior omitido

data() {
  return {

    foto: new Foto(),

    id: this.$route.params.id
  },
  // código posterior omitido
```

É através de `$route.params` que temos acesso ao parâmetro passado para o componente via rota. No caso, como o nome do nosso parâmetro é `id`, usamos `$route.params.id`.

Agora que já sabemos o ID da foto, podemos buscá-la em nossa API. Mas primeiro, precisamos criar o método `busca` em `FotoService`:

```
export default class FotoService {

  /* código anterior omitido */

  busca(id) {

    return this._resource
      .get({ id })
      .then(res => res.json());
  }
}
```

O método `busca` recebe o ID da foto e nos devolve uma promise que os fornecerá o objeto foto trazido de nossa API.

Agora, podemos usar este novo método em `Cadastro`, mas dentro do método `created`. Inclusive, precisamos testar se `this.id` possui valor, porque quando acessamos `Cadastro` diretamente sem passar parâmetro, o `id` será `undefined`.

```
// alurapic/src/components/cadastro/Cadastro.vue

// código anterior omitido
```

```
created() {  
  
  this.service = new FotoService(this.$resource);  
  
  if(this.id) {  
    this.service  
      .busca(this.id)  
      .then(foto => this.foto = foto);  
  }  
}  
// código posterior omitido
```

Perfeito! Um teste demonstra que ao clicarmos em qualquer foto em `Home`, somos direcionados para `Cadastro` com o formulário exibido com os dados da foto que desejamos alterar.