



03

Erros no servidor e Evolução do Vuex

Transcrição

Ao retornarmos para a experiência do usuário, precisaremos exibir uma mensagem de erro caso a combinação de e-mail e senha não estejam corretos.

Atualmente quando inserimos uma combinação errada no formulário conseguimos visualizar no console o retorno 401 do backend. Isso significa que o usuário não possui permissão para realizar a requisição desejada.

Em `login.vue`, dentro do método `efetuarLogin()`. Despacharemos o evento de efetuar login, e dentro de `.then()` redirecionamentos o usuário para a página de gerentes e além disso realizaremos um `catch()` no caso de erro enviaremos uma mensagem.

```
<script>
export default {
  data() {
    return {
      usuario: {},
      mensagemErro: ""
    };
  },
  methods: {
    efetuarLogin() {
      this.$store
        .dispatch("efetuarLogin", this.usuario)
        .then(() => this.$router.push({ name: "gerentes" }))
        .catch(err => {
          if (err.request.status === 401) {
```

```
        this.mensagemErro = "Login ou senha  
inválido(s)!!!";  
    }  
    });  
  }  
}  
};  
</script>
```

[COPIAR CÓDIGO](#)

Dentro do nosso template, criaremos, com o auxilio do bootstrap um alerta. Mas essa mensagem só será exibida sobre uma condicional.

```
<p class="alert alert-danger" v-if="mensagemErro">{{  
mensagemErro }}</p>  
    <button type="submit" class="btn btn-primary btn-  
block">Logar</button>
```

[COPIAR CÓDIGO](#)

Feito isso, ao inserirmos uma senha errada no formulário, é exibida a mensagem de erro que configuramos. Caso as informações estejam corretas, a mensagem de erro não será exibida.

Voltaremos a atenção para a estrutura do nosso projeto. Criamos um arquivo chamado `provedor.js`. O padrão do Vuex é chamar este arquivo de `store`, então modificaremos essa nomeação.

Teremos erros em todos os trechos em que realizamos a importação, então precisaremos localizar esses problemas e corrigi-los.

Feito isso, tudo opera normalmente.

