01

A classe NegociacaoController

Transcrição

Começando deste ponto? Você pode fazer o download (https://github.com/alura-cursos/javascript-avancado-i/archive/aula3.zip) completo do projeto até aqui e continuar seus estudos.

Nós já temos a negociação criada, agora, precisamos capturar as ações do usuário e interagir com o modelo por meio de uma `controller` no modelo MVC. Nós criaremos instâncias de negociação. Primeiramente, precisamos criar o `controller`.

Atualmente, no `index.html`, só temos a importação do arquivo `Negociacao.js`:

```
<script src="js/app/models/Negociacao.js"></script>
```

Lembrando que não temos mais a outra tag `<script>` usada nos testes. Caso você ainda a tenha, deve apagá-la.

Em seguida, criaremos a classe `NegociacaoController.js`. Usamos a convenção usada anteriormente de começar o nome da classe em caixa alta. Quando clicarmos no botão "Incluir" do formulário, vamos submetê-lo e depois, chamaremos uma ação do `controller` para adicionar uma negociação na tabela. Por enquanto, focaremos na criação da `Negociacao` com base nas informações dos formulários.

Usaremos o método `adiciona()` na classe `NegociacaoController`. O método receberá como parâmetro um `event`.

```
class NegociacaoController {

  adiciona(event) {
        event.preventDefault();
        alert('Chamei ação no controller');
  }
}
```

Chamamos o `event.preventDefault()`. Quando submetemos o formulário, se não cancelamos o comportamento padrão do mesmo, ele será recarregado. Com o `event.preventDefault()`, a `controller` cancelará a submissão do formulário para poder capturar os dados da negociação e incluir na lista. Ainda não colocamos `constructor`, porque a negociação ainda não tem nenhum atributo de classe.

Como faremos a ligação do método `adiciona()` com a submissão do formulário? Precisaremos fazer a importação do `NegociacaoController.js`. Também teremos uma tag `<script>` com a qual iremos instanciar esta `controller`.

```
<script src="js/app/models/Negociacao.js"></script>
<script src="js/app/controllers/NegociacaoController.js"></script>
<script>
    let negociacaoController = new NegociacaoController();
</script>
```

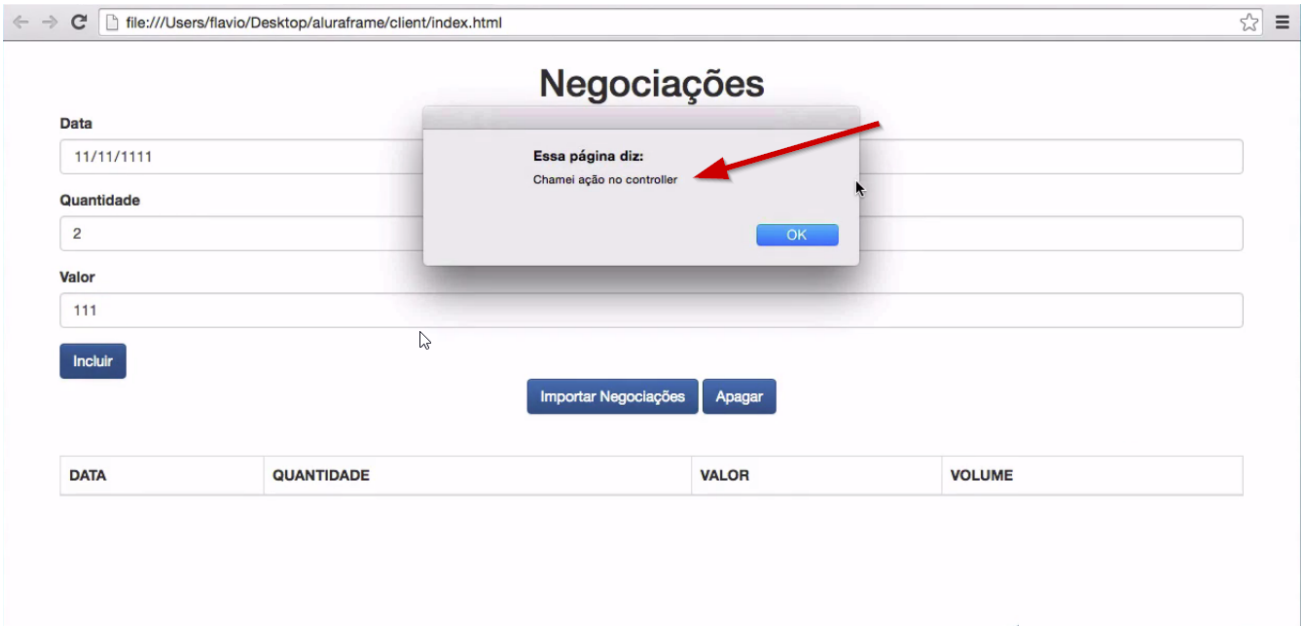
Observe que fizemos declaração da variável `negociacaoController` com o `let`, conforme convencionamos. Agora, nossa `controller` já está acessível na página inteira.

Em seguida, dentro da tag `<form>`, adicionaremos `onsubmit` para submeter o formulário. E na instância de `negociacaoController`, chamaremos o método `adiciona(event)`:

```
<body class="container">

    <h1 class="text-center">Negociações</h1>

        <form class="form" onsubmit="negociacaoController.adiciona(event)">
            <div class="form-group">
                <label for="data">Data</label>
                <input type="date" id="data" class="form-control" required autofocus/>
            </div>

    //...
```

A estratégia utilizada aqui é um pouco diferente do que vem sendo feita, geralmente, a associação entre tag e o evento é feita no parte JS do projeto. Mas inspirado por framework como o AngularJS, por exemplo, que faz a associação de uma ação da `controller` utilizando um evento. Usaremos isso, para escrever menos código na parte JavaScript. Como nossa página é dependente de JS, podemos fazer isso facilmente.

Se atualizarmos a página e preenchermos o formulário com dados aleatórios, após submetermos as informações, veremos a mensagem do `alert`.



Então, a ação da `controller` foi chamada. Mas o nosso objetivo não é exibir a mensagem, nós queremos criar um negociação. Teremos que gerar o código que instância a negociação. Faremos isto mais adiante.