

Você lembrou de atualizar a view? Nem eu!

Transcrição

Seja bem-vindo!

Começando deste ponto? Você pode fazer o [download \(https://github.com/alura-cursos/javascript-avancado-i/archive/projetoCompleto.zip\)](https://github.com/alura-cursos/javascript-avancado-i/archive/projetoCompleto.zip) completo do projeto da primeira parte do curso e continuar seus estudos a partir daqui.

Nossa aplicação já funciona e totaliza o volume no fim da tabela. Mas ainda falta implementar uma regra de negócio no modelo: vamos adicionar a funcionalidade que nos permite esvaziar uma lista.

Atualmente, o arquivo `ListaNegociacoes.js` está assim:

```
class ListaNegociacoes {  
  
  constructor() {  
  
    this._negociacoes = [];  
  }  
  
  adiciona(negociacoes) {  
  
    this._negociacoes.push(negociacao);  
  }  
  
  get negociacoes() {  
  
    return [].concat(this._negociacoes);  
  }  
}
```

Adicionaremos o método `esvazia()` abaixo da linha do `return()` :

```
esvazia() {  
  
  this._negociacoes = [];  
}
```

O `array` de negociações receberá uma nova lista e com isso, apagará todos os itens da anterior. Ao clicarmos no botão "Apagar" do formulário, queremos que as informações sejam apagadas da lista de negociações. Para isto, sabemos que quem atua no modelo é a `controller` mediante às ações do usuário. Em seguida, no arquivo `NegociacaoController.js`, adicionaremos o método `apaga()` :

```
apaga() {  
  
  this._listaNegociacoes.esvazia();  
}
```

```
this._negociacoesView.update(this._listaNegociacoes);

this._mensagem.texto = 'Negociações apagadas com sucesso';
this._mensagemView.update(this._mensagem);
}
```

O método solicitará o modelo `_listaNegociacoes.esvazia()`. Observe que adicionamos o `update()`. Quando atualizarmos a View, esta será recarregada automaticamente e a tabela ficará limpa.

Temos que associar a página a partir do evento de clique, e chamar o `apaga()`. Para isto, no `index.html`, adicionaremos o `onclick()` dentro da tag `<button>` de "Apagar":

```
<div class="text-center">
  <button class="btn btn-primary text-center" type="button">
    Importar Negociações
  </button>
  <button onclick="negociacaoController.apaga()" class="btn btn-primary text-center" type="button">
    Apagar
  </button>
</div>
```

Se recarregarmos a página no navegador, nenhum problema será apontado no console e conseguiremos cadastrar uma negociação normalmente. Quando clicarmos em "Apagar", a tabela ficará vazia. A View foi atualizada com os dados do modelo da lista de negociações atualizada e a mensagem foi exibida.

Sempre que atualizarmos o modelo, é possível esquecer de chamar o modelo da View? Sim. Nós temos duas Views: `MensagemView` e `NegociacoesView`. É preciso atualizar o modelo nas duas e, depois, chamar o `update()`. Mas um sistema maior pode conter um número muito superior de Views e corremos o risco de esquecer de adicionar o método. Esta é uma grande responsabilidade para o desenvolvedor.

Será que existe alguma maneira de automatizarmos a atualização da View? Vamos alterar a lista de negociações nos métodos `adiciona()` e `apaga()`, quando criarmos a instância de negociações no `constructor()`. Tentaremos encontrar uma solução para que a View seja atualizada automaticamente assim que o modelo for alterado.

