

Resumindo

Transcrição

Vamos revisar o que vimos até aqui e entender o quanto evoluímos. Nós criamos a primeira `Controller` da negociação: `NegociacaoController`. Vimos como associar uma ação do usuário, como a submissão do formulário, e chamar o método `controller`. Para criar um negociação do DOM, tivemos que criar os elementos do DOM, equivalente ao `input` da `quantidade`, da `data` e do `valor` para capturar os valores, sendo possível depois, instanciar uma negociação. Comentamos também como não é recomendável fazer isto todas as vezes que chamarmos o método `adiciona()`. Por isso, colocamos como propriedade da classe `NegociacaoController` o elemento do DOM.

Para evitarmos escrever repetidas vezes `document.querySelector`, nós usamos o "truque" de colocá-lo na variável `$`. Mas vimos que neste processo, o `querySelector` perdia o contexto do `document`, e o `this` deixava de apontar para este. Para resolver o assunto, usamos a chamada para o método `bind()` e o `$` - equivalente ao `querySelector` - fizesse uma referência para o `document`. Fizemos um sintaxe parecida com jQuery.

Falamos também que não era suficiente capturarmos a data do formulário e passá-la como parâmetro para o construtor de `Date`, porque o `input` vinha no formato **ano, mês e dia**. Depois, tivemos que fazer algumas transformações e vimos que o `Date` aceita trabalhar com alguns parâmetros. Passamos um `array` para o `Date`, também passamos uma `string` com ano, mês e dia, cada item separado por uma vírgula. Existem várias formas de trabalharmos com a data, mas precisamos ajustar a data que lemos no formulário e está num formato `string`, para o formato compatível para o construtor do `Date`.

Eu deixei um desafio para você: eu queria que fosse usado um construtor que usasse **ano, mês e dia**, cada um disposto como um parâmetro diferente no construtor. Porém, neste formato, o mês não pode ser passado de `1` a `12`, no caso, precisaria ser de `0` a `11`. então, foi necessário realizar algumas transformações e precisamos um pouco de programação funcional.