

 05

Resumindo

Transcrição

Toda vez que um modelo for atualizado, tanto por nós quanto pela aplicação, você se lembrará de chamar o `update()` na View? É improvável.

Vimos como tirar a responsabilidade do desenvolvedor e colocar a chamada da atualização da View no código, quando o modelo for alterado.

Seguindo o exemplo do `listaNegociacoes`, em que os métodos `adiciona()` e `esvazia()` eram chamados, será esse o momento no qual vamos disparar a atualização da View. Optamos por colocar "armadilhas", funções passadas para o construtor da classe que são chamadas sempre que os métodos `adiciona()` ou `esvazia()` forem usados. Estas funções recebem um código que acessa a View da `controller` e executará o método `update()`.

Mas a solução fracassou, porque a `function()` era enviada para o construtor do modelo que é a armadilha. Quando isso acontecia, o `this` era dinâmico, ou seja, não pertencia a `controller`, e sim, ao `model`. Então, tentamos acessar no `this` a nossa View. Vimos como fazer isto por meio da API de reflexão do JavaScript, **Reflection API**, usando `reflect.apply()`. Ela recebe o nome do método, o contexto em que queremos executar o método, além dos parâmetros que serão passados para o método para corrigir o `this` no momento da execução da função. Mas vimos que este processo não era necessário. Em vez disso, usamos uma *arrow function*.

As *arrow functions* possuem um escopo léxico, enquanto as funções padrões têm um escopo dinâmico. Isto significa que, se temos uma função em JavaScript que varia de acordo com o contexto no qual é chamada, o `this` léxico de uma *arrow function* manterá o mesmo `this` em todas as chamadas da função. Isso torna o código mais enxuto, porque não precisarmos passar o contexto do construtor do modelo.

Nos exercícios, você terá a chance de praticar o conteúdo trabalhado, mas vale ressaltar que poluir o modelo com esse código de infraestrutura não é uma boa ideia. Atacaremos tal questão na próxima seção do curso, e descobriremos como disparar atualizações na View sem colocar "armadilhas". No entanto, os conceitos abordados aqui já devem estar bem consolidados. Vamos praticar!