

Podcast

Disciplina: Frameworks front end

Título do tema: Angular: utilizando o TypeScript com o Framework da Google.

Autoria: Anderson da Silva Marcolino

Leitura crítica: Paulo Henrique Santini

Abertura:

Olá, ouvinte! No *podcast* de hoje vamos falar um pouco sobre o *two-way-data binding* ou ligação de dados bidirecional, muito utilizado no Angular e outros *frameworks front end* baseados em JavaScript.

Para aqueles que estão familiarizados com o JavaScript, as informações da linguagem são inseridas nos elementos HTML de diversos modos, porém sempre com a condição de que esta alteração seja realizada por meio da alteração do modelo de documento de objeto, o conhecido DOM. Para isso cria-se o que chamamos em inglês de *handlers* ou, em português, os manipuladores de DOM. Contudo, quando passamos a adotar frameworks ou arcabouços como o Angular, o processo de passagem de informações do JavaScript para o HTML ocorre de modo facilitado! É aí que entra a ligação de dados bidirecional ou o *two-way-data binding*. Este mecanismo permite que a passagem de dados manipulados pelo JavaScript ocorra automaticamente. E como estamos falando de uma estrada de mão dupla, ou seja, bidirecional, os dados são alterados do JS para o DOM e o conteúdo do DOM altera também os dados do JS. Podemos imaginar a visão e o modelo de uma página web. Enquanto a visão é desenvolvida considerando o HTML, CSS etc. o modelo é desenvolvido considerando JS. Caso haja mudanças no modelo, a visão é atualizada, já se houver mudanças na visão, o modelo é então atualizado. A maneira mais simples de se implementar isso é criando, por exemplo, um elemento que possua a combinação de dois colchetes e internamente, dois parênteses. Enquanto os colchetes fazem a ligação das propriedades, os parênteses fazem a ligação com o evento. Assim, o que foi modificado o DOM ou no JS, será alterado sem a necessidade de uma atualização da página, como é feito quando pressionamos F5 do teclado. A atualização automática de informações traz dinamicidade até então nunca vista em projetos web, já que, além de não necessitar de atualização da página, permite a exibição de informações mais rapidamente. Integrando tais possibilidades com um *back end* bem desenvolvido, dados inseridos por outros usuários no mesmo projeto podem ser atualizados dinamicamente. A ligação de dados bidirecional também pode utilizar elementos visuais e implicar em mudanças de estilização ou de imagens, já que pode-se passar para tais valores na interface, um endereço de um recurso específico. Não confunda atualizações automáticas por meio das folhas e estilos. A ligação

bidirecional de dados busca as atualizações de elementos nos dados e interações dos usuários, geralmente alterados à nível de servidor, ou seja, utilizando os dados salvos em bancos de dados relacionais ou não relacionais. É como se imaginássemos um programa desenvolvido para desktop, que tem sua interface atualizada automaticamente quando determinadas ações são realizadas, deste modo também se comportam as aplicações web, tudo devido ao *two-way-data binding*! Indo além, frameworks como o React podem ser inseridos no Electron, um invólucro que permite que uma aplicação totalmente web seja apresentada como uma aplicação puramente desktop. Obviamente, por se tratar de uma aplicação web, faz-se necessário manter sempre uma conexão com a Internet. Como exemplos de tais aplicações temos o WhastApp web e também o Spotify. Recomendo que busque mais sobre o two-way data binding na documentação oficial do Angular e os pratique em seus projetos, bem como sobre o Electron, uma boa opção para transformar projetos web em aplicações desktop robustas e atraentes.

Fechamento:

Este foi nosso *podcast* de hoje! Até uma próxima oportunidade e bons estudos!