



Primeiro contato com data binding

Transcrição

Antes de prosseguirmos, nos atentemos ao título que aparece na página da aplicação: "Welcome to app!". Ao voltarmos ao nosso template, e procurarmos o template do componente que sabemos que é carregado, `app.component.html`, veremos `Welcome to {{ title }}`!. No entanto, não é "title" que lemos na tela, e sim, "app".

O Angular se tornou famoso na época em que foi lançado por causa de um recurso chamado **Data binding**, ou "associação de dados". Assim como em `app.component.ts` há `title = 'app'`, existe uma propriedade no componente, no caso chamada `title`. A fonte de dados é a propriedade do nosso componente, e seu valor é `app`.

O *Data binding* implica em uma associação de dados com uma fonte de dados que, no nosso caso, está no componente, com seu template (nomenclatura do Angular), ou *view*. Nele, quando encontramos esta sintaxe chamada de **Angular expression** (AE), e quando o Angular for renderizar este template do componente, ele se deparará com uma lacuna, que neste caso está apontando para a propriedade `title`.

Esta propriedade existe em `app.component.ts`, então o Angular acessará seu valor e o jogará no template. Isso é interessante pois para realizarmos uma mudança dessas em JavaScript, tradicionalmente, precisamos manipular o DOM, realizando o `document.querySelector()`, selecionando o elemento e mudando o `textContent`.

A ideia do Angular com *Data binding* é justamente evitar perda de tempo manipulando DOM, e que possamos fazer algo que realmente vá agregar ao nosso cliente. Sendo assim, se trocarmos `app` por `alurapic` em `app.component.ts`, e salvarmos o projeto, sabemos que o novo valor para o `template` será outro.

Teríamos que recarregar a página no navegador, porém, ao abrirmos a página, a mudança já estará feita. Esta é uma vantagem fantástica do Angular CLI: qualquer alteração na aplicação é feita instantaneamente no navegador. Isso é perfeito quando se trabalha com dois monitores, pois em um vamos desenvolvendo, e no outro vemos as modificações sendo feitas.

Para quem já utilizou ferramentas *browser sync* (sincronização de navegador) para automatizar tarefas de front end, o Angular CLI servirá muito bem. Neste momento inicial, é importante entendermos que o uso de duas chaves abertas e duas fechadas representam uma *Angular Expression*. Toda vez que o programa encontrar algo nesta sintaxe, será buscado seu conteúdo, sua **expressão**, nas propriedades do componente. Isto é *Data binding*, que funciona como se o `template` sugasse os dados do componente.

Vamos mudar a aplicação, e começar a adequá-la para que ela realmente se torne uma aplicação de gerenciamento de imagens?