Ciclo de vida de um componente no angular 9

Quando um componente é inicializado, destruído ou possui alguma de suas propriedades de Input alteradas, temos alguns eventos especiais do angular que são automaticamente disparados, esses eventos sçao chamados de lifecycle hook ou, em português, gatilhos do ciclo de vida.

Existem basicamente, 8 destes métodos especias que são chamados por motivos e em momentos específicos e nesse vídeos abordaremos exatamente isso, trataremos do quando e por quê cada um destes métodos é disparado, todos estes eventos serão, aqui apresentados em sua devida ordem de execução, a começar pelo OnChanges.

O **OnChanges** é disparado sempre que o valor de um property-binding é atualizado e também antes da execução do OnInit.

**OnInit** é disparado quando o componente é inicializado, é aqui que faremos as primeira atribuições e também invocação de métodos que devem ser executados durante a inicialização do componente.

**DoCheck (verifique)** é executado a cada ciclo de verificação de mudanças do componente, ou seja, sempre que o componente sofrer mudanças ocorrerá essa execução.

**AfterContentInit** depois que o conteúdo do componente for inicializado, ou seja, após inserir conteúdos externos na view e isso tbm vale para property-binding.

**AfterContentChecked** depois que o conteúdo for checado ()

**AfterViewChecked** a cada verificação de conteúdo e isto vale tanto para conteúdo o pai, quanto para conteúdo filho

**Destroy** será executado antes da diretiva/componente ser destruído e removido da view

Os ciclos de vida e quando são disparados



AfterViewInit - é chamado apenas durante a primeira alteração ciclo de detecção, Você pode usá-lo se precisar saber quando o primeiro ciclo de detecção de alterações é executado.



Agora seguiremos para a parte prática

**OnChanges**

* O componente pai vincula dados a propriedade de entrado do filho.
* O componente recebe esses dados através do input e o **OnChanges** é disparado**.**
* Após 5 segundos a função de **setTimeout** retorna e o valor de arbitraryData (que está vinculado ao componente filho) é alterado, deste modo o valor alterado flui para o componente filho e novamente a função **OnChanges** é disparada.

**OnInit**

* É acionado uma única vez após a inicialização da Input Properties
* O componente pai vincula dados a propriedade de entrado do filho.
* O componente recebe esses dados através do input e dispara o **OnChanges** após isso o **OnInit** é disparado fazendo com que lifecycleTicks seja incrementado e após os 5 segundo o **setTimeout** retorna alterando o valor da propriedades arbitraryData, mas nesse momento **OnInit** não é disparado.

**DoCheck**

* É acionado a cada ciclo de detecção do angular, qualquer ação executada aqui ocorrerá sempre que um ciclo de alteração for disparado.
* Basicamente permite que os desenvolvedores verifiquem seus dados manualmente
* **\* Este evento requer muito cuidado uma vez que será executado a cada ciclo de alteração pode impactar na performance do componente**

Vale lembrar que estas interfaces são implementadas por boas práticas, mas que ou sem a implementação destas interfaces, o angular verificará se há na classe algum método que corresponda a algum destes ciclos de vida e caso exista os executará automaticamente.