Angular-8

Angular Component Life Cycle Hooks

Presentation BY RAJ PRUDHVI

Contact us:https://training.uplatz.com
Email: info@uplatz.com
Phone: +44 7836 212635

Uplatz

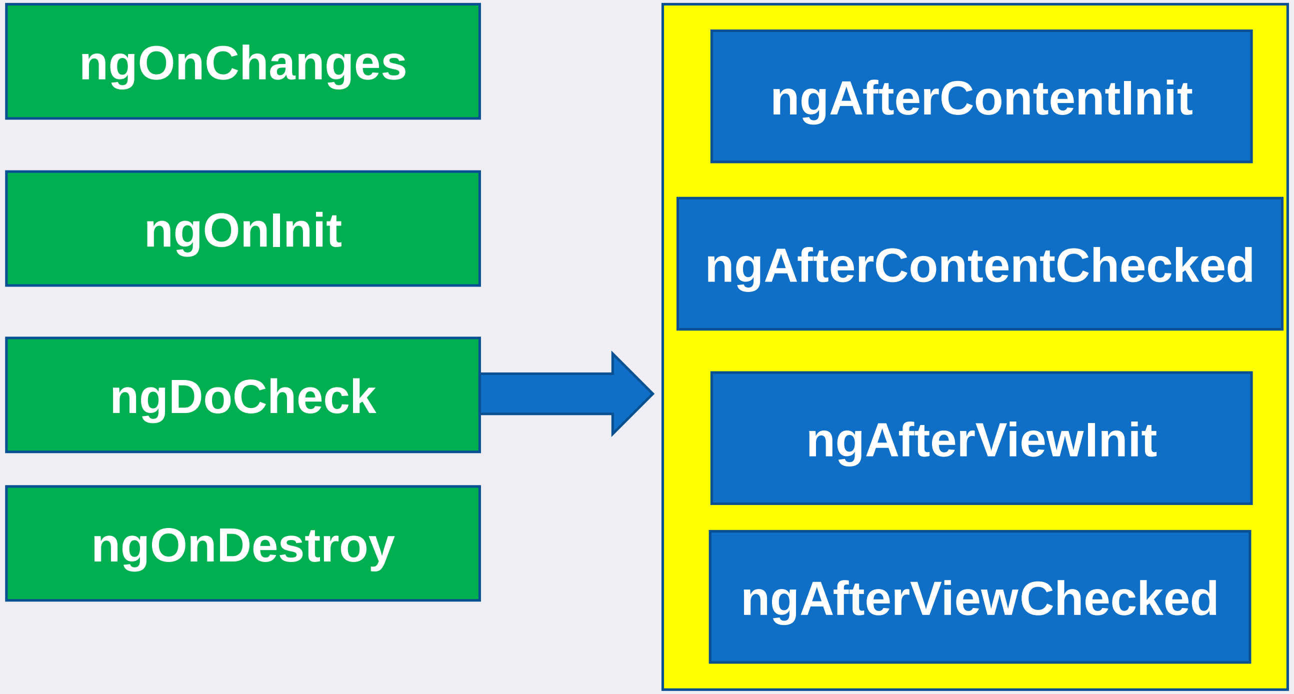
Angular component lifecycle hooks

A component has a lifecycle managed by Angular

- ➢ Creates the component
- ➢ Renders the component
- ➢ Creates and renders the component children
- ➢ Checks when the component data-bound properties change
- ➢ Destroys the component before removing it from the DOM

Angular component lifecycle hooks

To tap into and react when these life cycle events occur, angular offers several lifecycle hooks as shown in the image below.



Angular component lifecycle hooks

The 3 most commonly used hooks are

Life Cycle Hook	Purpose
ngOnChanges	Executes, every time the value of an input property changes. The hook method receives a SimpleChanges object containing current and previous property values. This is called before ngOnInit
ngOnInit	Executes after the constructor and after ngOnChanges hook for the first time. It is most commonly used for component initialization and retrieving data from a database
ngOnDestroy	Executes just before angular destroys the component and generally used for performing cleanup

Angular component lifecycle hooks

There are 3 simple steps to use the Life Cycle Hooks

Step 1 : Import the Life Cycle Hook interface. For example, to use **ngOnInit( )** life cycle hook, import OnInit interface.

```
import { OnInit } from '@angular/core';
```

Step 2 : Make the component class implement the Life Cycle Hook interface, using
the implements keyword as shown below. This step is optional, but good to have so you will get editor support and flags errors at compile time if you incorrectly implement the interface method or make any typographical errors.

```
export class SimpleComponent implements OnInit { }
```

Angular component lifecycle hooks

Step 3 : Write the implementation code for the life cycle interface method. Each interface has a single hook method whose name is the interface name prefixed with ng.

```
ngOnInit()
{
    console.log('OnInit Life Cycle Hook');
}
```

Angular ngOnChanges lifecycle hook

Let's understand ngOnChanges life cycle hook with a simple example. Here is what we want to do. As soon as the user starts typing into the text box, we want to capture the current and previous value and log it to the browser console . We can very easily achieve this by using the ngOnChanges life cycle hook.

ngOnChanges, is called every time the value of an input property of a component changes. So first let's create a SimpleComponent with an input property

Angular ngOnInit lifecycle hook

ngOnInit

Executes after the constructor and after ngOnChanges hook for the first time. It is most commonly used for component initialization and retrieving data from a database

Angular ngOnDestroy lifecycle hook

ngOnDestroy

Executes just before angular destroys the component and generally used for performing cleanup

Thank You

Contact us:https://training.uplatz.com

Email: info@uplatz.com

Phone: +44 7836 212635