**What are Life Cycle Hooks?**

Lifecycle hooks are special methods angular calls during the creation , update and destruction of components and directives.

Implementing them lets you run custom logic at various stages

|  |  |  |
| --- | --- | --- |
| Hooks | | Called When |
| ngOnChanges() | | Input properties changes |
| ngOnInit() | | Component initialized |
| ngDoCheck() | | Changes detection runs |
| ngAfterContentInit() | | Content Projected into Component |
| ngAfterContentChecked() | | Projected Content checked |
| ngAfterViewInit() | | Component’s Views Checked |
| ngAfterViewChecked() | | Component’s views checked |
| ngOnDestroy() | Component is about to be Destroyed | |

Create Component: user-card

* ng g user-card

user-card.component.html

<div class="card">

    <h3> User card</h3>

    <p> Name: {{name}}</p>

</div>

user-card.component.ts file

import { AfterContentChecked, AfterContentInit, AfterViewChecked, AfterViewInit, Component, DoCheck, Input, OnChanges, OnDestroy, OnInit, SimpleChanges } from '@angular/core';

@Component({

  selector: 'app-user-card',

  templateUrl: './user-card.component.html',

  styleUrls: ['./user-card.component.css']

})

export class UserCardComponent implements

OnChanges,OnInit,DoCheck,

AfterContentInit,AfterContentChecked,

AfterViewInit,AfterViewChecked,OnDestroy

{

  @Input() name:string='';

  private intervalId:any;

  constructor(){

    console.log('[Constructor] component instance Created');

  }

  ngOnChanges(changes: SimpleChanges): void {

    console.log('[ngOnChanges] input changed:',changes);

  }

   ngOnInit(): void {

    console.log('[ngOnInit] Initialization logic here');

    this.intervalId= setInterval(()=>{

      console.log('[Interval] simulating background works...')

    },2000);

  }

  ngAfterViewChecked(): void {

    console.log('[ngAfterViewChecked] content Checked!')

  }

  ngAfterViewInit(): void {

    console.log('[ngAfterViewInit] View initilized!')

  }

  ngAfterContentChecked(): void {

    console.log('[ngAfterContentChecked] View checked!')

  }

  ngAfterContentInit(): void {

    console.log('[ngAfterContentInit]View Initialized!')

  }

  ngDoCheck(): void {

    console.log('[ngDoCheck]Custom change detection logic running!')

  }

  ngOnDestroy(): void {

     console.log('[ngOnDestroy]component being destroyed!')

     clearInterval(this.intervalId);

  }

}

App.component.html

<div class="container">

  <button class="btn btn-primary" (click)="toggleUser()"> Toggle User Card</button>

<button class="btn btn-info" (click)="changeName()"> ChangeName</button>

</div>

<app-user-card \*ngIf="showUser" [name]="userName"></app-user-card>

App.component.ts file

import { Component } from '@angular/core';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  title = 'HttpService';

  userName='Alice';

  showUser=true;

  toggleUser(){

    this.showUser= !this.showUser;

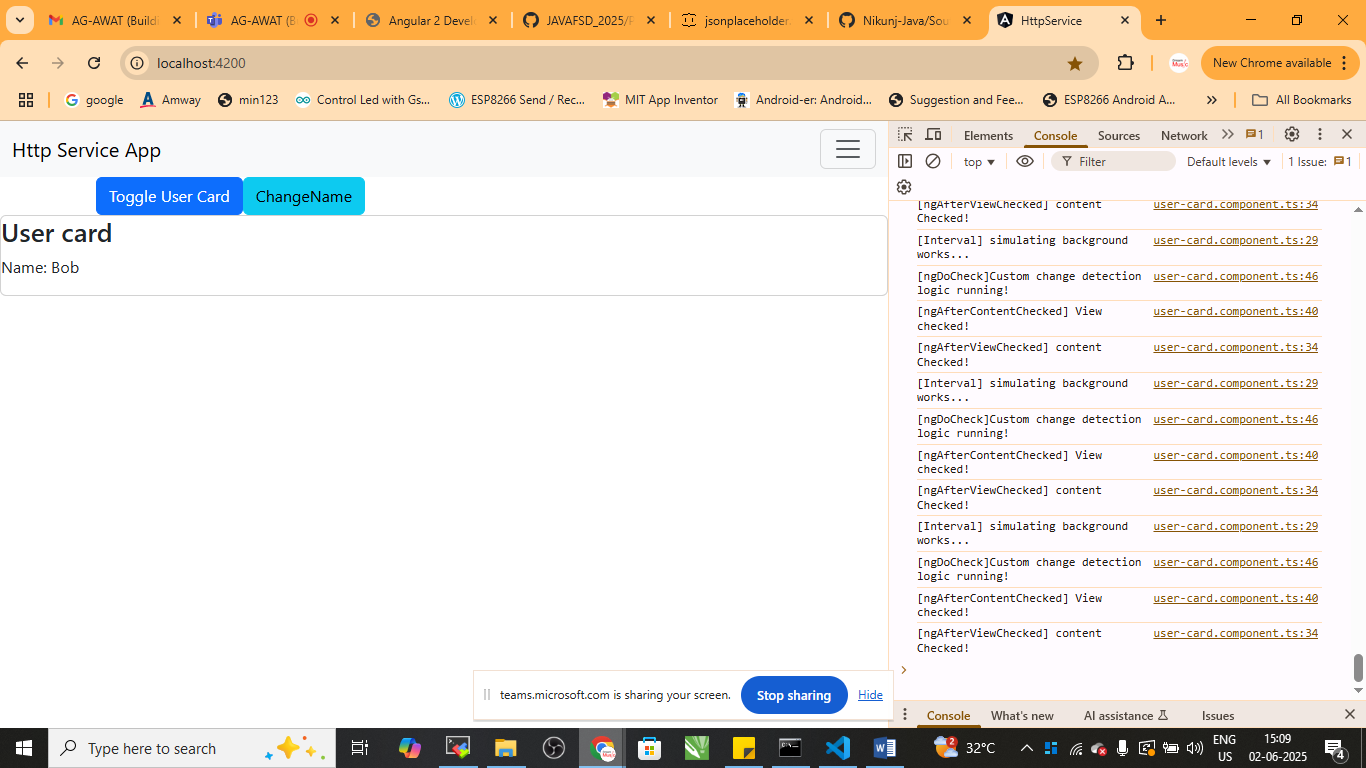
  }

  changeName(){

    this.userName= this.userName ==='Alice'?'Bob' : 'Alice';

  }

}



Rguht click 🡪 inspect 🡪 console