M&I SYSTEMS GROUP

VOJVODINA ICT CLUSTER

Angular - Lifecycle hooks

GO4Code

Marko Cvijanović
Mihailo Bubnjević

Component Lifecycle

- Svaka komponenta ima neki životni ciklus
- Lifecycle počinje kada Angular instancira klasu te komponente i renderuje njen view
- Lifecycle se nastavlja kada Angular proveri da li se neki povezani (data-bound) properti promenio
- Životni ciklus se završava kada Angular uništi komponentu
- Direktive imaju sličan lifecycle

Lifecycle hook metode

- **Lifecycle hooks** su specijalne metode kojima možemo da se “zakačimo (hook into)” na neki deo životnog ciklusa i izvršimo neki kod
- Svaka komponenta možemo da implementira jednu ili više lifecycle hook metoda
- Konstruktor nije lifecycle hook, on je funkcionalnost typescript klase
- Dobra praksa je da implementiramo interface odgovarajuće metode gde je ime interfejsa naziv lifecycla-a bez “ng”

Lifecycle hook metode

- **ngOnInit**
- Ova metoda će biti pozvana kada se ta komponenta potpuno inicijalizuje
- Za svaku komponentu biće pozvana samo jednom

Lifecycle hook metode

- **ngOnDestroy**
- Poziva se kada se komponenta uništi i izbaci iz DOM-a
- Ako treba da obavimo neko ‘čišćenje’ koristimo onDestroy

Lifecycle hook metode

- **ngOnChanges**
- Poziva se kada @Input() iz roditeljske komponente promeni vrednost
- Poziva se pre ngOnInit
- Ako nemamo @Input() vrednosti koje prosleđujemo neće se desiti
- Nije pametno raditi skupe operacije u ngOnChanges

Lifecycle hook metode

- **ngDoCheck**
- Poziva se posle svakog poziva na **ngOnChanges()**
- Poziva se jednom posle **ngOnInit()**
- Poziva se na svaku promenu data-bound propertija
- Ne bi trebalo implementirati doCheck i onChanges u istoj komponenti

Lifecycle hook metode

- Svaka komponenta ima pristup sledećim lifecycle metodama
  - ngOnInit
  - ngOnChanges
  - ngDoCheck
  - ngOnDestroy
- Svaka komponenta koja je takođe i child neke druge komponente ima
  - ngAfterContentInit
  - ngAfterContentChecked
  - ngAfterViewInit
  - ngAfterViewChecked