

Metode

1. Uvod

Do sada smo u komponentama koristili promenljive i pomoću šablona (mehanizmom interpolacije) ugrađivali njihove vrednosti u web stranicu. Da bi se web stranica tokom svog prikazivanja mogla prilagođavati podacima i aktivnostima korisnika, potrebno je da sadrži određene funkcionalnosti. Kako bi to postigli, koristimo **metode** koje su ili ugrađene ili ih sami pišemo:

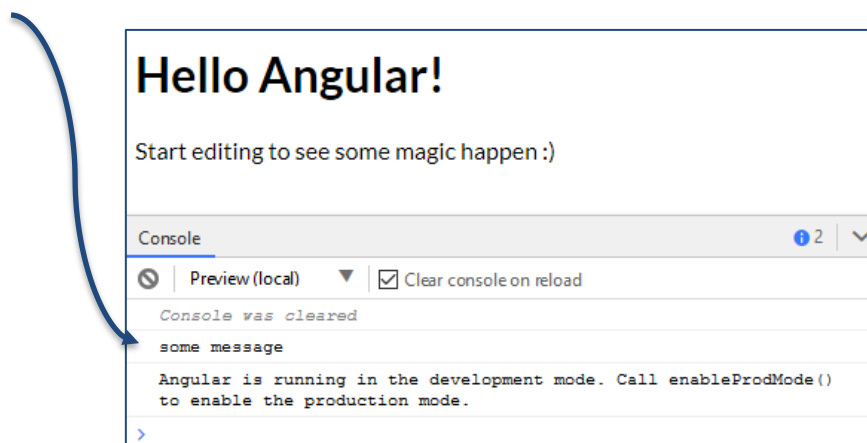
- Postojeće metode za ispis poruka korisniku
- Sopstvene metode (metode koje smo mi kreirali)
- Metode vezane za životni ciklus komponente
- Metode koje se pozivaju pri određenim aktivnostima korisnika

2. Ispis poruka

Radi lakšeg testiranja i debug-ovanja korisno je ispisivati pomoćne poruke u toku izvršavanja programa. Postoje dva načina ispisa poruka:

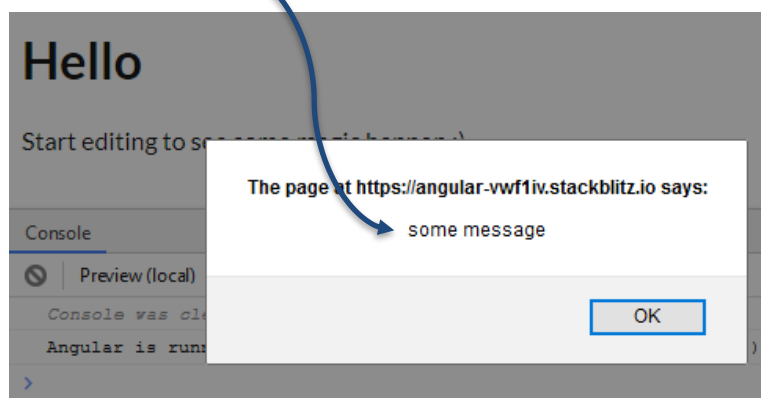
1. U konzoli

`console.log('some message');`



2. Putem alert dijaloga

`alert('some message');`



3. Metode u Angular komponentama

Pored promenljivih, komponente mogu sadržati i metode koje, kada su pozvane, izvršavaju neki deo programskog koda.

Metode se sastoje od:

- Naziva
- Parametara (opciono)
- Tela

Na slici je prikazana metoda bez parametara i metode sa parametrima (1 ili više).

```
myMethod1() {  
  console.log('Method without parameters!');  
}  
  
myMethod2(parameter1) {  
  console.log('Method with one parameter.');}  
  
myMethod3(parameter1, parameter2) {  
  console.log('Method with two parameters.');}
```

Pozivi metoda se dešavaju nastupanjem nekih događaja. Neki od tih događaja nastaju prilikom učitavanja komponente, a neki tokom interakcije sa korisnikom.

! Za sve metode kojima dajemo proizvoljan naziv **moramo obezbediti i njihovo pozivanje** (ili se one nikada neće izvršiti).

4. Životni ciklus Angular komponente

Životni ciklus komponente obuhvata sve događaje od njenog **instanciranja** (zauzimanja memorije) do njenog **uništenja** (oslobađanja memorije).

4.1 Instanciranje

Prvi događaj u životnom ciklusu komponente je njeno instanciranje. Angular vrši instanciranje komponente kada u HTML-u web stranice naiđe na tagove koji odgovaraju selektoru te komponente (definisano svojstvom *selector* u *@Component* dekoratoru).

Prilikom instanciranja će se izvršiti metoda **constructor()** ukoliko je definisana unutar komponente. Ovom metodom se najčešće omogućava povezivanje s drugim elementima aplikacije (dependency injection). O tome ćemo pričati kasnije.

! Konstruktor se izvršava na samom početku kreiranja komponente. U tom trenutku još uvek ne postoje svi delovi komponente, pa bilo kakve **složenije operacije ne bi trebalo pozivati iz konstruktora**.

4.2 Lifecycle hook methods

Nakon instanciranja, u životnom ciklusu Angular komponente je moguće obraditi i druge događaje. Za svaki od tih događaja postoji rezervisan naziv metode. Neke od tih metoda su:

- `ngOnInit` – poziva se kada se završi kreiranje komponente
- `ngOnChanges` – poziva se kada dođe do promene na ulaznoj vrednosti iz roditeljske komponente
- `ngOnDestroy` – poziva se pre uništavanja i brisanja komponente iz memorije

* Metoda koju ćemo koristiti na ovom kursu je **ngOnInit**, a više o ostalim metodama možete pročitati u Angular dokumentaciji, ukoliko želite.



Angular će izvršiti metode **constructor** i **ngOnInit** zahvaljujući njihovom specijalnom nazivu. Ako ih nazovemo drugačije, one se **neće izvršiti po automatizmu**.

5. Događaji i rukovanje događajima

Angular omogućava obradu događaja koji nastaju u interakciji korisnika i web stranice. Angular razlikuje sledeće događaje:

- | | | | | |
|------------|-------------|----------------|--------------|--------------|
| • (focus) | • (cut) | • (keypress) | • (mouseup) | • (dragover) |
| • (blur) | • (copy) | • (keyup) | • (click) | • (drop) |
| • (submit) | • (paste) | • (mouseenter) | • (dblclick) | |
| • (scroll) | • (keydown) | • (mousedown) | • (drag) | |

Ovi događaji služe kao "okidači" za povezivanje metoda koje smo implementirali u komponenti. Da bi se događaj obradio, potrebno je **uz konkretan HTML element** naznačiti **vrstu događaja** koju očekujemo i **naziv metode** koju tada treba pozvati:

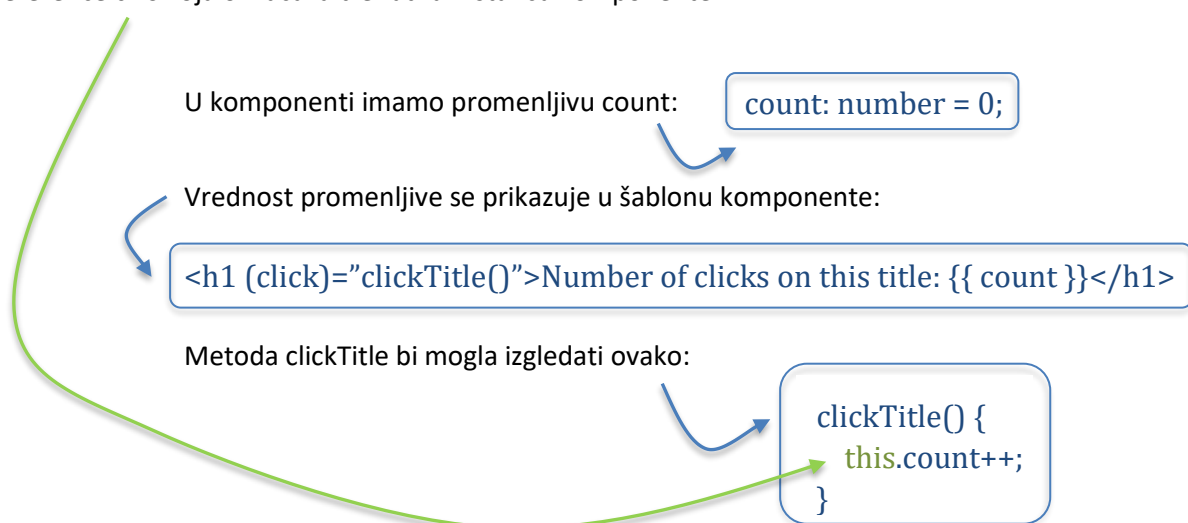
```
<h1 (click)="clickTitle()">Some title</h1>
```

Implementacija metode clickTitle u telu komponente bi mogla izgledati ovako:

```
clickTitle() {  
  console.log("You clicked the title!");  
}
```

5.1 Pristup promenljivama iz metoda

U telu metode moguće je pristupati vrednostima promenljivih. Promenljive se koriste uz navođenje reference **this** koja označava trenutnu instancu komponente.



Zadaci

Alat koji će se koristiti za pravljenje Angular aplikacija na vežbama je <https://stackblitz.com/>

1. Kreirati 3 dugmeta sa nazivima omiljenih pesama. Klikom na neko dugme omogućiti ispis osnovnih informacija o pesmi (naziv, izvođač, link do pesme na YouTube-u).

Domaći

1. Kreirati kviz sa pitanjima uz koja se nalaze po dva dugmeta sa ponuđenim odgovorima. Klikom na dugme, u paragrafu ispod pitanja ispisati da li je odgovor tačan ili nije. Primeri pitanja:
 - Koji web framework je prvi nastao: AngularJS / Angular
 - Angular je web framework za: Front-end / Back-end
 - U implementaciji Angular komponenti se koristi jezik: JavaScript / TypeScript
 - Web browser-i omogućavaju izvršavanje jezika: JavaScript / TypeScript
 - Tokom rada, brži odziv imaju aplikacije koje su: single page / multi page

Literatura

1. <https://angular.io/>
2. <https://material.angular.io/>
3. Materijali sa predavanja