

Povezivanje (binding) podataka komponente i šablona

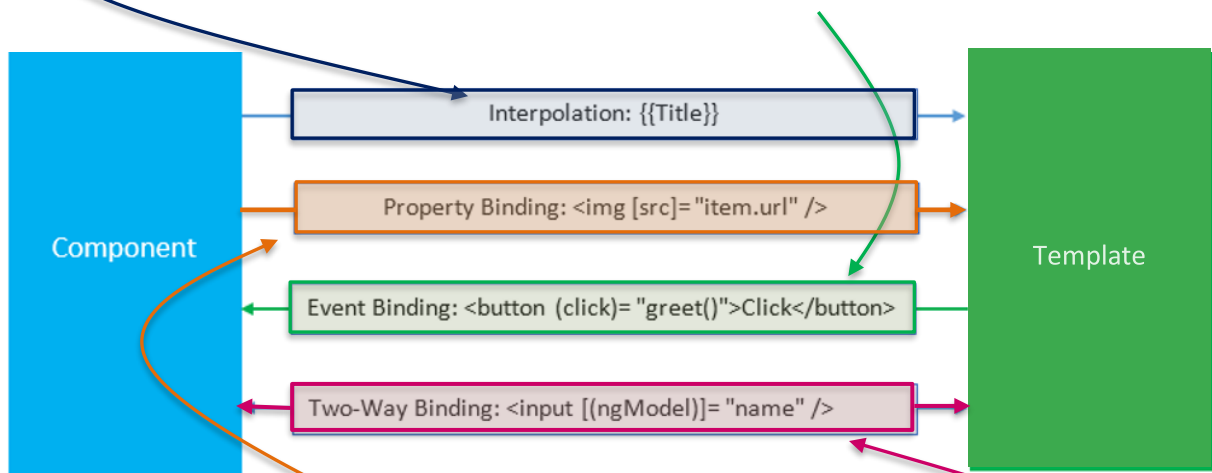
1. Povezivanje (binding) podataka

Angular omogućava sinhronizaciju vrednosti u modelu podataka sa prikazom tih vrednosti na web stranici. Ta usklađenost podataka u modelu sa stanjem web stranice je ključna da bi se obezbedila adekvatna interakcija sa korisnikom tokom životnog ciklusa stranice.

Angular omogućava povezivanje u različitim smerovima. U zavisnosti od željenih rezultata, potrebno je odabrati način povezivanja između šablona i komponente.

Sa dva načina prenošenja informacija smo se već susreli:

1. Od komponente ka šablonu (jednosmerno) - interpolacija
 - npr. u slučaju interpolacije `{{ Title }}` će davati ažurnu vrednost
2. Od šablona ka komponenti (jednosmerno) – povezivanje događaja (*event binding*)
 - npr. u slučaju događaja `<button (click)="greet()">Click</button>`



Pored ta dva načina, postoje još dva:

1. Od komponente ka šablonu (jednosmerno) – povezivanje svojstava (*property binding*)
 - npr. ``
2. Dvosmerno povezivanje
 - npr. `<input [(ngModel)]= "name"/>`

2. Komponenta → šablon

U ovom poglavlju će biti detaljnije obrađeno povezivanje podataka od komponente ka šablonu.

2.1 Interpolacija

Interpolaciju smo već koristili i ona je tipičan primer za smer komponenta → šablon. Vrednosti kojima raspolaže šablon su vezane za vrednosti koje postoje u komponenti.

- ! **Bilo kakva promena vrednosti u komponenti** će se **istovremeno** odraziti na prikaz tih vrednosti putem šablona.

2.2 Povezivanje svojstava – *property binding*

Osim interpolacije, povezivanje svojstava takođe ima smer komponenta → šablon. Ovaj način se koristi kada neki atribut HTML elementa treba povezati sa nekom vrednošću iz komponente. Povezivanje se vrši tako što se u šablonu odgovarajući atribut navodi unutar uglastih zagrada [].

- ! Kao i kod interpolacije, **bilo kakva promena vrednosti u komponenti** će se **istovremeno** odraziti na prikaz.

2.3 Razlika između interpolacije i povezivanja svojstava

Ako pogledamo primer ispod, primetićemo da će oba načina povezivanja dovesti do istog rezultata.

interpolacija

```
<a href="{{url}}">Link</a>
```

povezivanje
svojstava

```
<a [href]="url">Link</a>
```

Dakle, tekstualne podatke je u većini slučajeva moguće ugraditi u šablon interpolacijom.

- ! **Ako bi vrednost bila logičkog tipa, interpolacija ne bi funkcionisala.** Interpolacijom bi logičke vrednosti (*true*, *false*) postali **stringovi** 'true' i 'false'.

Pošto se u HTML-u logički atributi zapisuju drugačije (npr. disabled, selected, checked) *preporučljivo* je da se *za svojstva uvek koristi povezivanje svojstava*. Tada će Angular voditi računa o ispravnom nazivu atributa.

2.4 Povezivanje svojstava primenjeno na CSS stilove

Povezivanjem svojstava je moguće kontrolisati pridruživanje stilova nekom HTML elementu. Kao uslov za pridruživanje može poslužiti izraz čija istinitost određuje da li će stil biti pridružen ili ne.

Primer: postojeća CSS klasa se može primeniti na neki element ako se u njega ugradi atribut `[class.class_name]` pa mu se kao vrednost dodeli izraz koji određuje da li će stil biti primenjen.

weather.component.ts

```
@Component({
  selector: 'weather',
  templateUrl: './weather.component.html',
  styleUrls: ['./weather.component.css']
})
export class WeatherComponent {
  temperature: number = 25;
}
```

weather.component.css

```
.warm {
  color: red;
}
```

weather.component.html

```
<p [class.warm]="temperature">20</p>
```

Navedeni primer bi rezultovao sa: `<p class="warm">25</p>`

3. Šablon → komponenta

U ovom poglavlju će biti obrađeno povezivanje podataka od šablona ka komponenti.

3.1 Povezivanje događaja – *event binding*

Sa povezivanjem događaja smo se već susreli i koristili ga za obradu događaja koji se dešavaju na web stranici. Nastupanjem konkretnog događaja se poziva funkcija unutar komponente.

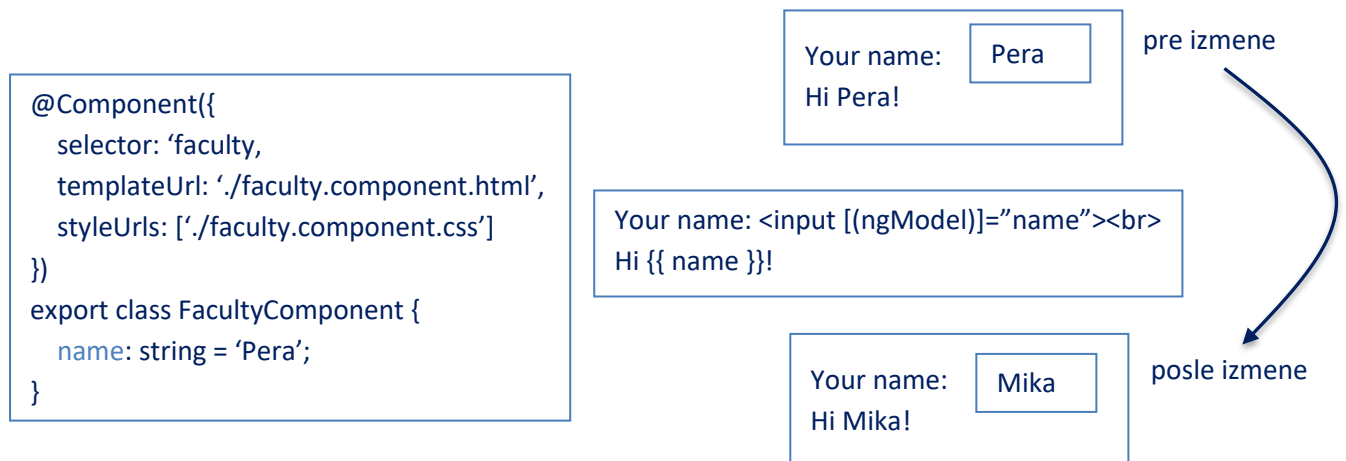
```
@Component({
  selector: 'faculty',
  templateUrl: './faculty.component.html',
  styleUrls: ['./faculty.component.css']
})
export class FacultyComponent {
  onClick() {
    console.log('Hello!');
  }
}
```

```
<button (click)="onClick()">Click here!</button>
```

Kada se klikne dugme, izvršiće se metoda *onClick* i u konzoli će se ispisati 'Hello!'.

4. Dvosmerno povezivanje – *two-way binding*

Povezivanje u oba smera je korisno kada korisniku treba prikazati neki podatak i istovremeno mu omogućiti da taj podatak menja. Dvosmerni binding predstavlja neku vrstu kombinacije povezivanja svojstava i povezivanja događaja, pa se rezervisana reč **ngModel** navodi kao **[(ngModel)]**. Ovaj tip povezivanja se obično koristi kod `<input>` elemenata.



Zadaci

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

1. Kreirati polje za unos iznosa u evrima i polje za unos iznosa u dinarima. Dodati i dva dugmeta tako da se jednim vrši konverzija iz dinara u evre, a drugim iz evra u dinare.

Domaći

1. Kreirati tri polja za unos tako da čine jednačinu, pa dodati dugme za proveru koje će ispod jednačine ispisati da li je jednakost tačna ili netačna. Na primer:
$$\text{cinilac1} \times \text{cinilac2} = \text{proizvod}$$

Literatura

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