

Angular databinding

Data Binding

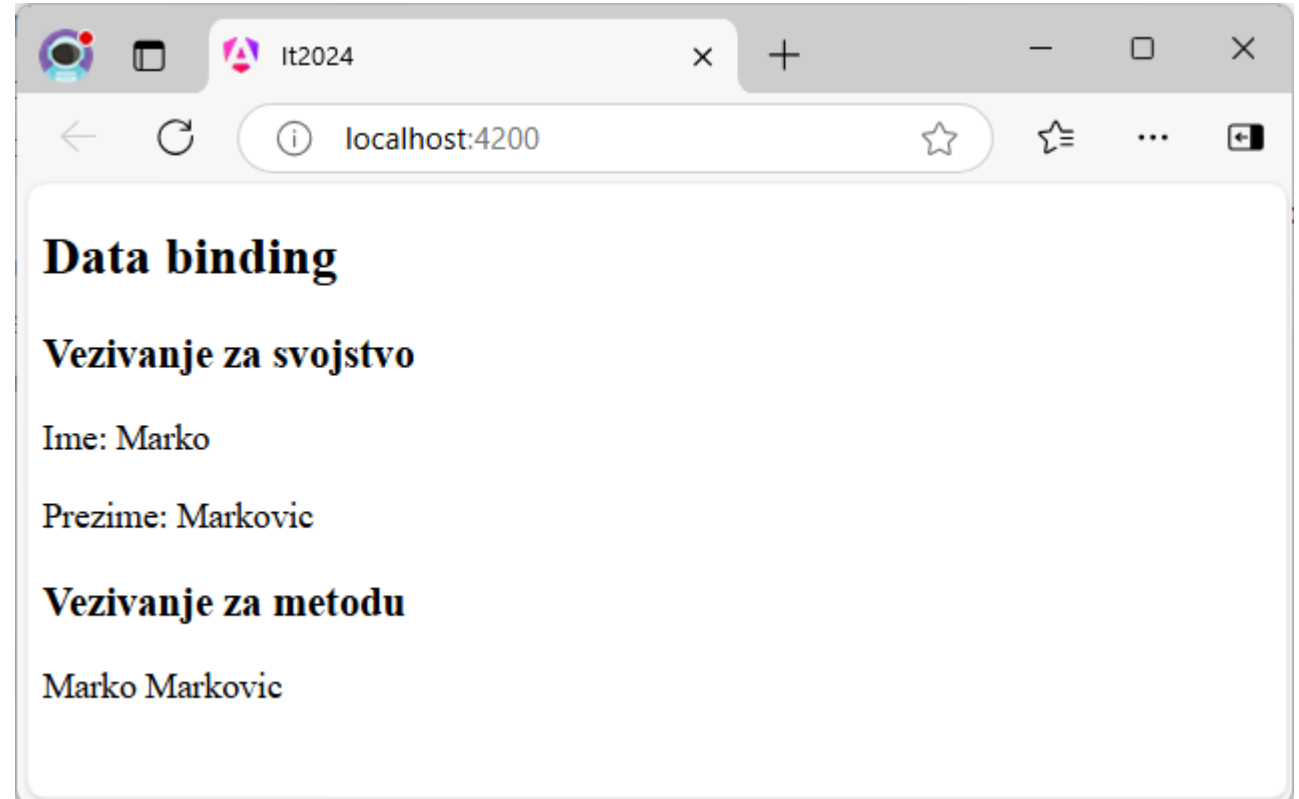
- Data Binding je proces sinhronizacije podataka između klase komponente i pogleda komponente
- Vršiti se povezivanje DOM elemenata sa svojstvima klase komponente
- Postoji 4 načina povezivanja podataka u Angularu:
 - Interpolacija
 - Povezivanje svojstva (Property binding)
 - Povezivanje događaja (Event binding)
 - Dvosmerno povezivanje (Two Way Binding)

Interpolacija -klasa komponente

```
export class AppComponent {  
  naslov = 'Data binding';  
  osoba = {ime: 'Marko', prezime: 'Markovic', grad: 'Beograd'};  
  
  citajIme(): string {  
    return this.osoba.ime + ' ' + this.osoba.prezime;  
  }  
}
```

Interpolacija-pogled

```
<h2>{{ naslov }}</h2>  
<h3>Vezivanje za svojstvo</h3>  
<p>Ime: {{ osoba.ime }}</p>  
<p>Prezime: {{ osoba.prezime }}</p>  
<h3>Vezivanje za metodu</h3>  
<p>{{ citajIme() }}</p>
```



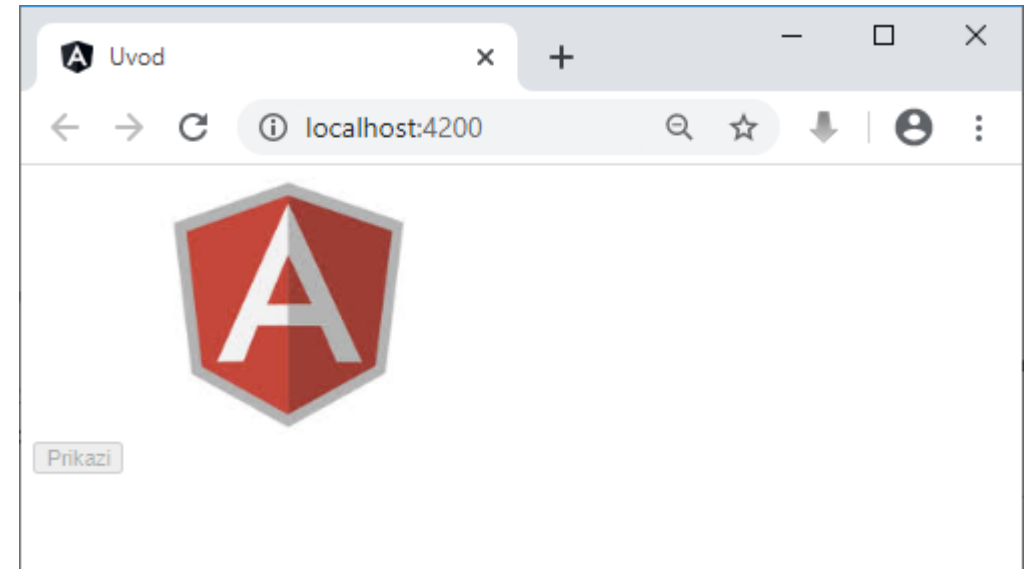
Property binding

- Jednosmerna komunikacija između klase komponente i pogleda
- Podaci se prosleđuju od klase komponente do pogleda komponente
- Koristi se kada želimo da postavite vrednost HTML svojstva (kao što su src, href, disabled, itd.) na osnovu podataka iz komponente
- Sintaksa koristi srednje zagrade [] oko HTML atributa kako bi se označilo da se vrednost vezuje za neko svojstvo komponente
- Svojstvo komponente za koje se vrši vezivanje se smešta unutar navodnika

Property binding

```
export class AppComponent {  
  adresaSlike = '/angular.png';  
  zabrani = true;  
}
```

```
<img [src]="adresaSlike">  
<br>  
<button [disabled]="zabrani">Prikazi</button>
```



Event binding

- Podaci se prosleđuju od pogleda komponente ka klasi komponente
- Izvršava akciju u komponenti kada korisnik izvršava akciju u korisničkom interfejsu pogleda npr. klikne na dugme u pogledu
- Ime događaja se stavlja unutar malih zagrada npr. (click)
- Zatim se događaju dodeljuje funkcija iz komponente

Event binding

klasa komponente:

```
export class AppComponent {  
  brojac = 0;  
  
  uvecaj(): void {  
    this.brojac++;  
    console.log('Vrednost brojaca: ', this.brojac);  
  }  
}
```

šablon komponente:

```
<button (click)="uvecaj()">Uvecaj</button>
```


Two Way Binding (dvosmerno povezivanje)

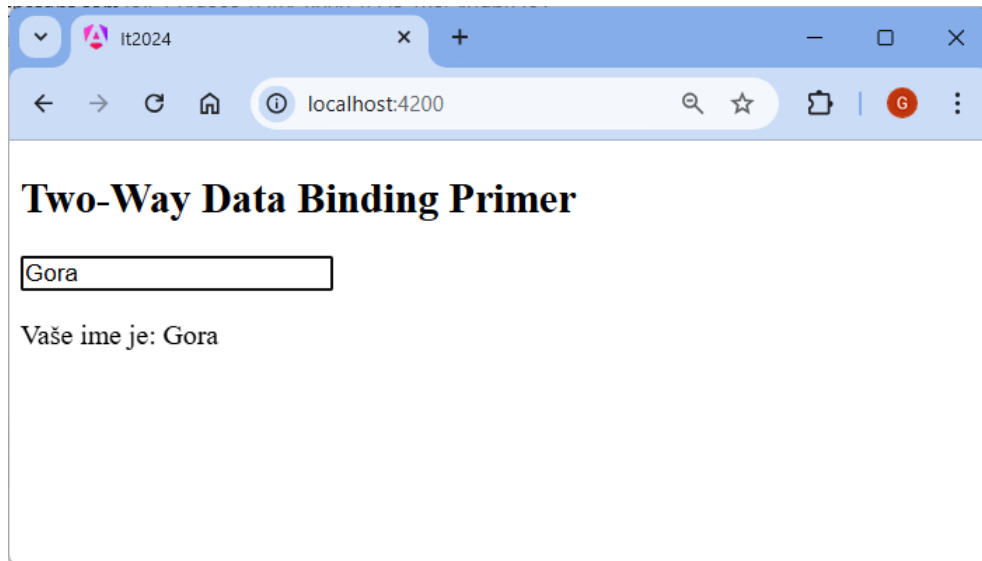
- Promene u klasi komponente se propagiraju do pogleda
- Promene u pogledu menjaju podatke u klasi komponente
- Dvosmerno povezivanje se koristi kod formi za unos podataka
- Angular koristi kombinaciju povezivanja sa svojstvom (od komponente ka pogledu) i povezivanja sa događajem (od pogleda ka komponenti) da bi ostvario dvosmerno povezivanje
- Koristi se ngModel direktiva
- Potrebno je da se u standalone komponentu importuje FormsModule da bi se koristila ngModel direktiva

Klasa komponente

```
import { Component } from '@angular/core';
import { FormsModule } from '@angular/forms';
@Component({
  selector: 'app-root',
  standalone: true,
  imports: [FormsModule],
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  ime= '';
}
```

Šablon komponente

```
<div>
  <h2>Two-Way Data Binding Primer</h2>
  <input [(ngModel)]="ime" placeholder="Unesite ime">
  <p>Vaše ime je: {{ ime }}</p>
</div>
```



Pitanje 1

Angular komponenta ima javno polje pod nazivom ime. Na koji način se sadržaj ovog polja prikazuje u šablonu komponente

- a. [ime]
- b. {{ ime }}
- c. { ime }

Odgovor: b

Pitanje 2

Angular komponenta ima javno polje: **sakrivanje** tipa boolean. Na koji način se svojstvo hidden nekog div elementa u šablonu vezuje za vrednost ovog polja?

- a. `<div [hidden]= "sakrivanje">Div 1</div>`
- b. `<div {hidden}= "sakrivanje">Div 1</div>`
- c. `<div {{hidden}}= "sakrivanje">Div 1</div>`

Odgovor: a

Pitanje 3

Klasa komponente ima funkciju prikazi() koja nema ulazne parameter i ne vraća vrednost. U šablonu komponente treba omogućiti poziv ove funkcije pri kliku na dugme:

- a. `<input type="button" value="Click" (click)="prikazi" />`
- b. `<input type="button" value="Click" [click]="prikazi()" />`
- c. `<input type="button" value="Click" (click)="prikazi()" />`

Odgovor: c

Pitanje 4

Definisano je polje klase komponente tipa string pod nazivom **ime**. Za dvosmerno povezivanje vrednosti tekstualnog polja šablona komponente sa poljem ime komponente koristimo izraz:

- a. `<input type="text" [(ngModel)]= "ime">`
- b. `<input type="text" ngModel="ime">`
- c. `<input type="text" (ngBind)="ime">`

Odgovor: a

Angular direktive

Referenciranje bootstrap biblioteke

index.html

```
<script src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.slim.min.js"
integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-Fy6S3B9q64WdZWQIiU+q4/2Lc9npb8tCaSX9FK7E8HnRr0Jz8D60P9d05Vg3Q9ct"
crossorigin="anonymous"></script>
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css"
integrity="sha384-xOo1HFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwNLD69Npy4HI+N"
crossorigin="anonymous">
```

Angular direktive

- Olakšavaju manipulaciju DOM elementima
- Ako se koristi standalone komponenta, potrebno je importovati CommonModule u komponentu da bi se koristile direktive
- Tri tipa direktiva:
 - Komponente
 - Strukturalne direktive dodaju ili uklanjaju DOM elemente
 - ngFor
 - ngSwitch
 - ngIf
 - Atributske direktive menjaju izgled ili ponašanje DOM elementa
 - ngModel
 - ngClass
 - ngStyle

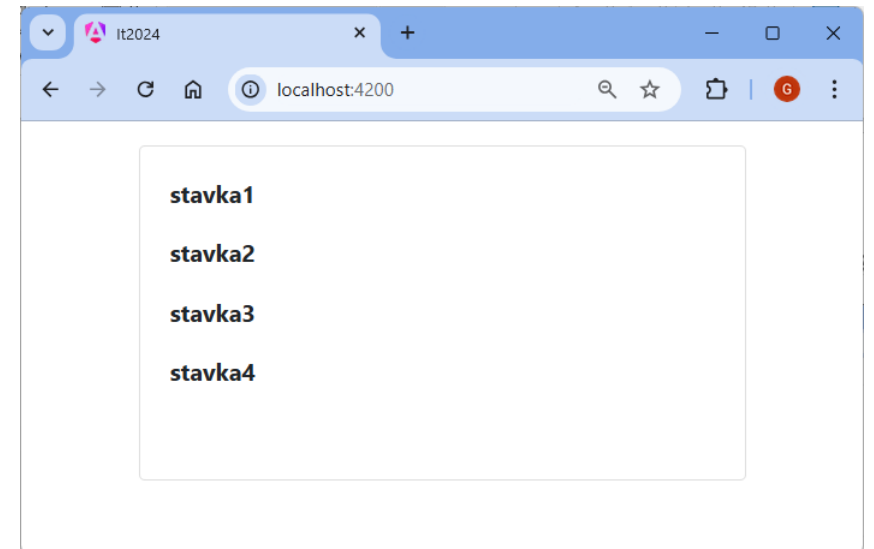
Uključivanje modula CommonModule

```
import { Component } from '@angular/core';
import { CommonModule } from '@angular/common';
@Component({
  selector: 'app-root',
  standalone: true,
  imports: [CommonModule],
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  stavke: string[] = ['stavka1', 'stavka2', 'stavka3', 'stavka4'];
}
```

Direktiva *ngFor

Omogućava kreiranje DOM elemenata za svaki član niza

```
<div class="container mt-3">
  <div class="card">
    <div class="card-body">
      <p *ngFor="let stavka of stavke">
        <strong>{{stavka}}</strong>
      </p>
      <br>
    </div>
  </div>
</div>
```



Direktiva *ngIf

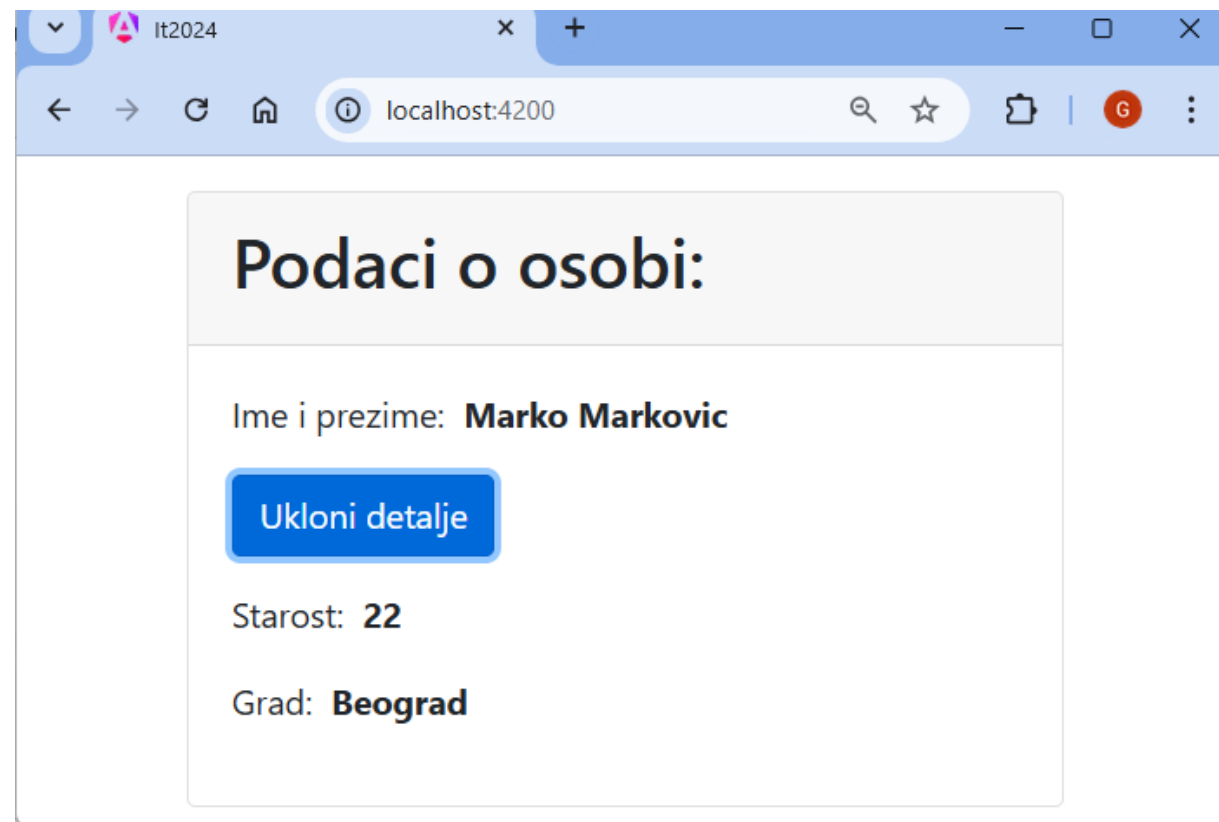
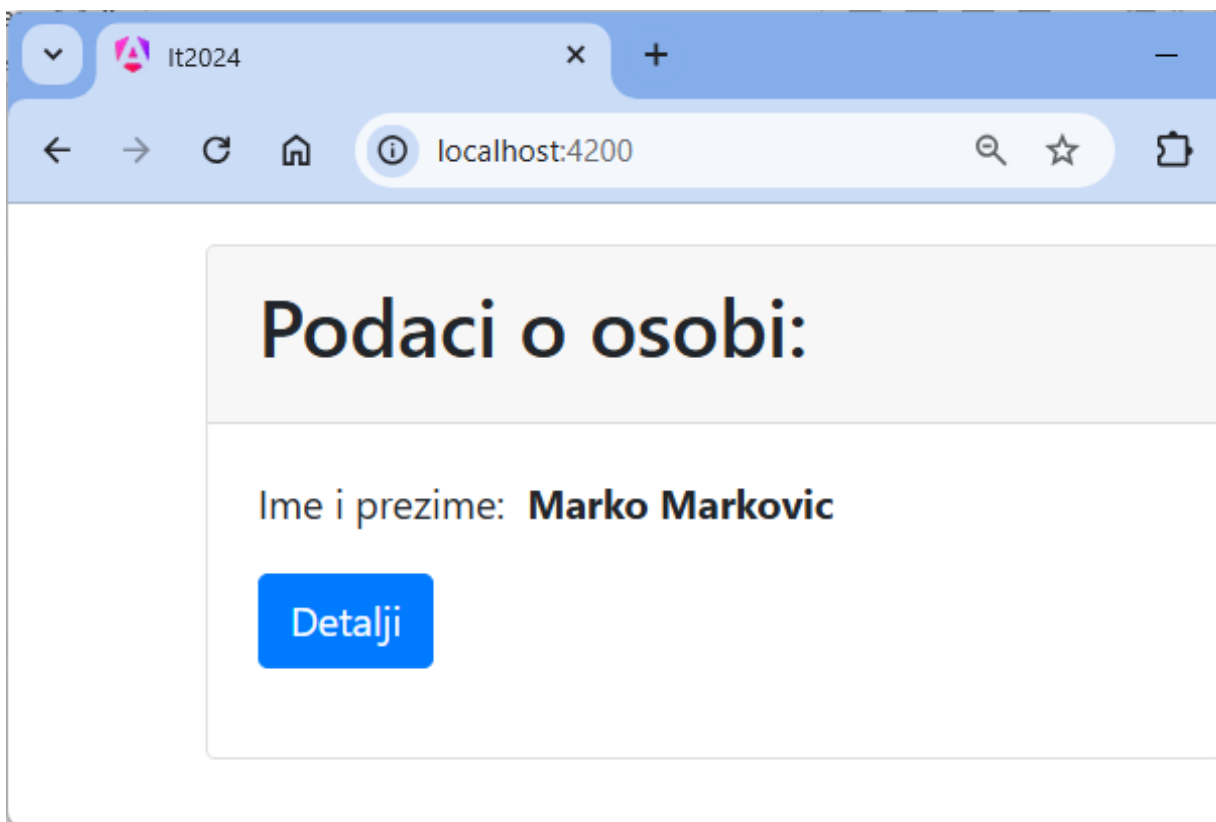
```
export class NgifComponent {  
  
  osoba = {ime:'Marko', prezime: 'Markovic', starost:22, grad:'Beograd'};  
  prikaziDetalje = false;  
  dugmeTekst = 'Detalji';  
  
  toggleDetalji() {  
    this.prikaziDetalje = !this.prikaziDetalje;  
    this.dugmeTekst = this.prikaziDetalje ? 'Ukloni detalje' : 'Detalji';  
  }  
}
```

Omogućava uslovno prikazivanje elementa.

Direktiva *ngIf – šablon komponente

```
<div class="container mt-3">
  <div class="card">
    <div class="card-header">
      <h2>Podaci o osobi:</h2>
    </div>
    <div class="card-body">
      <p>
        Ime i prezime:
        <strong>{{osoba.ime}} {{osoba.prezime}}</strong>
      </p>
      <button class="btn btn-primary mb-3" (click)="toggleDetalji()">{{dugmeTekst}}</button>
      <div *ngIf="prikaziDetalje">
        <p>
          Starost:
          <strong>{{osoba.starost}}</strong>
        </p>
        <p>
          Grad:
          <strong>{{osoba.grad}}</strong>
        </p>
      </div>
    </div>
  </div>
</div>
```

Direktiva *ngIf



Direktiva ngSwitch

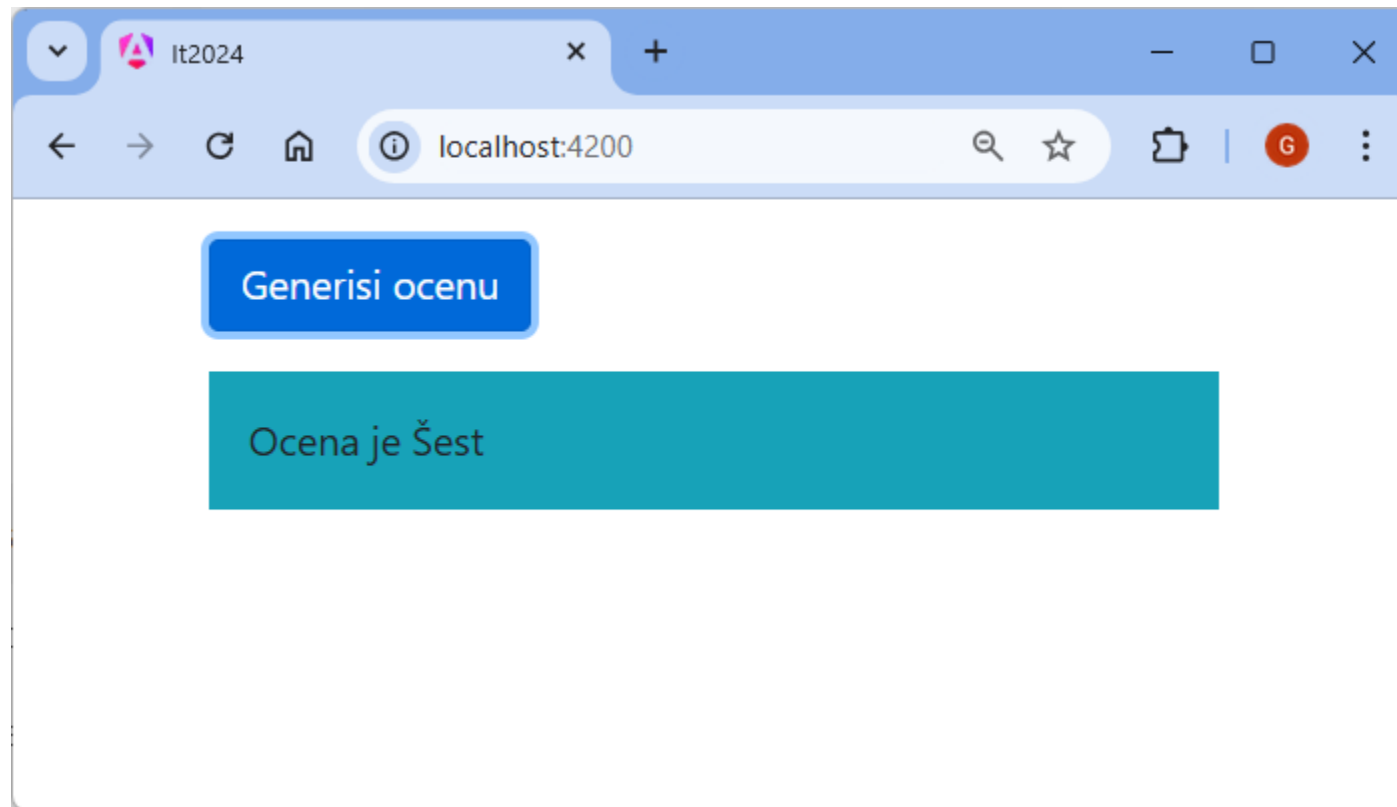
Omogućava prikaz različitih elemenata u zavisnosti od uslova

```
export class NgswitchComponent {  
  
  ocena : number = 1;  
  promeni() {  
    this.ocena = Math.floor(Math.random() * 6) + 5;  
  }  
  
}
```


Direktiva ngSwitch

```
<div class="container mt-3">
  <div class="mb-3">
    <button class="btn btn-primary" (click)="promeni()">Generisi
ocenu</button>
  </div>
  <div [ngSwitch]="ocena" class="bg-info p-3">
    <div *ngSwitchCase="5">Ocena je Pet</div>
    <div *ngSwitchCase="6">Ocena je Šest</div>
    <div *ngSwitchCase="7">Ocena je Sedam</div>
    <div *ngSwitchCase="8">Ocena je Osam</div>
    <div *ngSwitchCase="9">Ocena je Devet</div>
    <div *ngSwitchCase="10">Ocena je Deset</div>
    <div *ngSwitchDefault>Nepoznata ocena</div>
  </div>
</div>
```

Direktiva ngSwitch



Direktiva ngClass

NgClass je Angular direktiva koja omogućava dinamičko dodavanje ili uklanjanje CSS klasa na HTML elementima

```
export class NgclassComponent {  
  
    jeCrveno: boolean = true;  
    jePlavo: boolean = false;  
  
    promeniBoju() {  
        this.jeCrveno = !this.jeCrveno;  
        this.jePlavo = !this.jePlavo;  
    }  
}
```

Css fajl komponente

```
div{
  width: 400px;
  height: 100px;
  padding: 10px;
  margin-bottom: 10px;
}

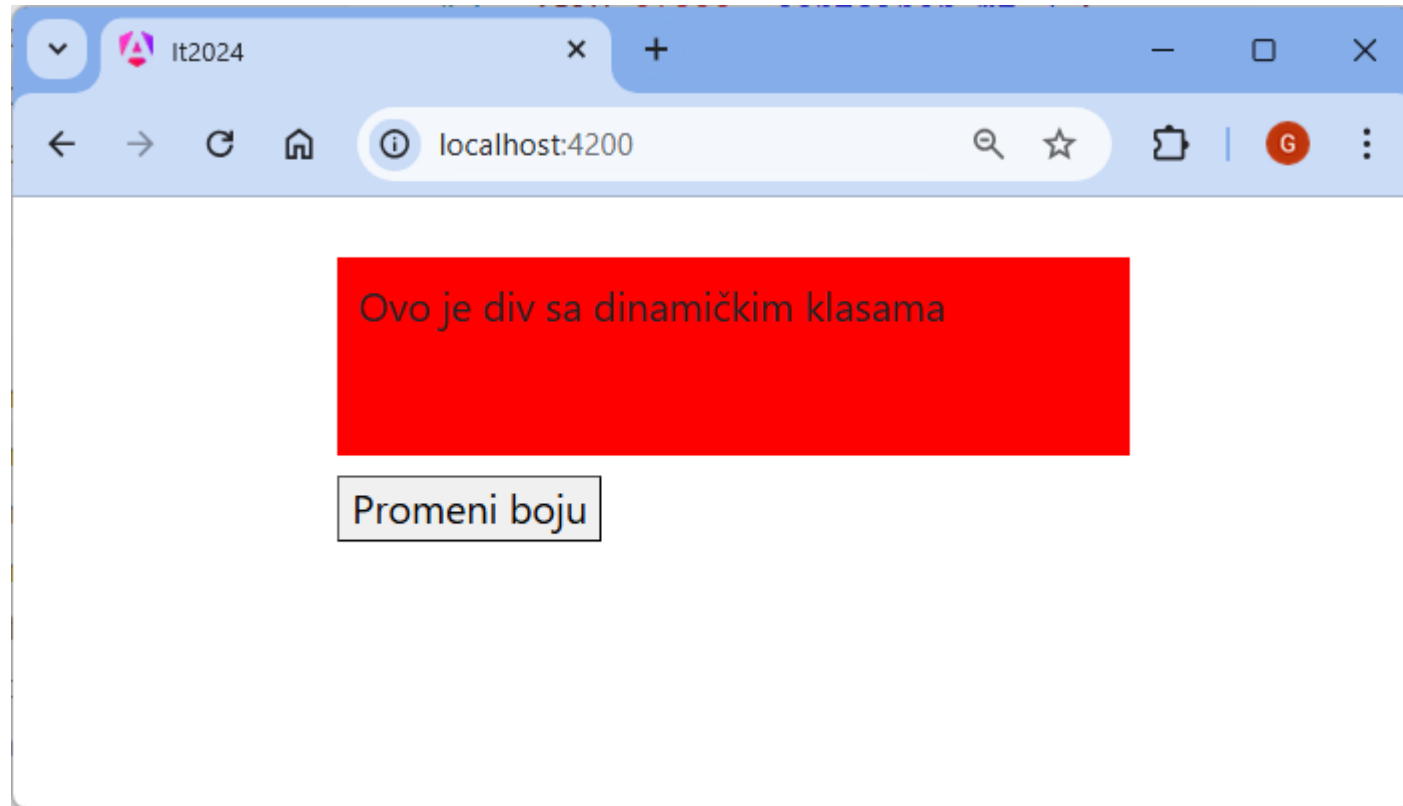
.crvena-boja {
  background-color: red;
}

.plava-boja {
  background-color: blue;
}
```

Šablon komponente

```
<div class="container mt-3">  
  <div [ngClass]="{ 'crvena-boja': jeCrveno, 'plava-boja': jePlavo }">  
    Ovo je div sa dinamičkim klasama  
  </div>  
<button (click)="promeniBoju()">Promeni boju</button>  
  
</div>
```

ngClass



Generisanje klase

ng g class osoba

```
export class Osoba {  
    constructor(public osobaId:number, public ime:string, public prezime:string) {  
    }  
}
```

Klasa AppComponent

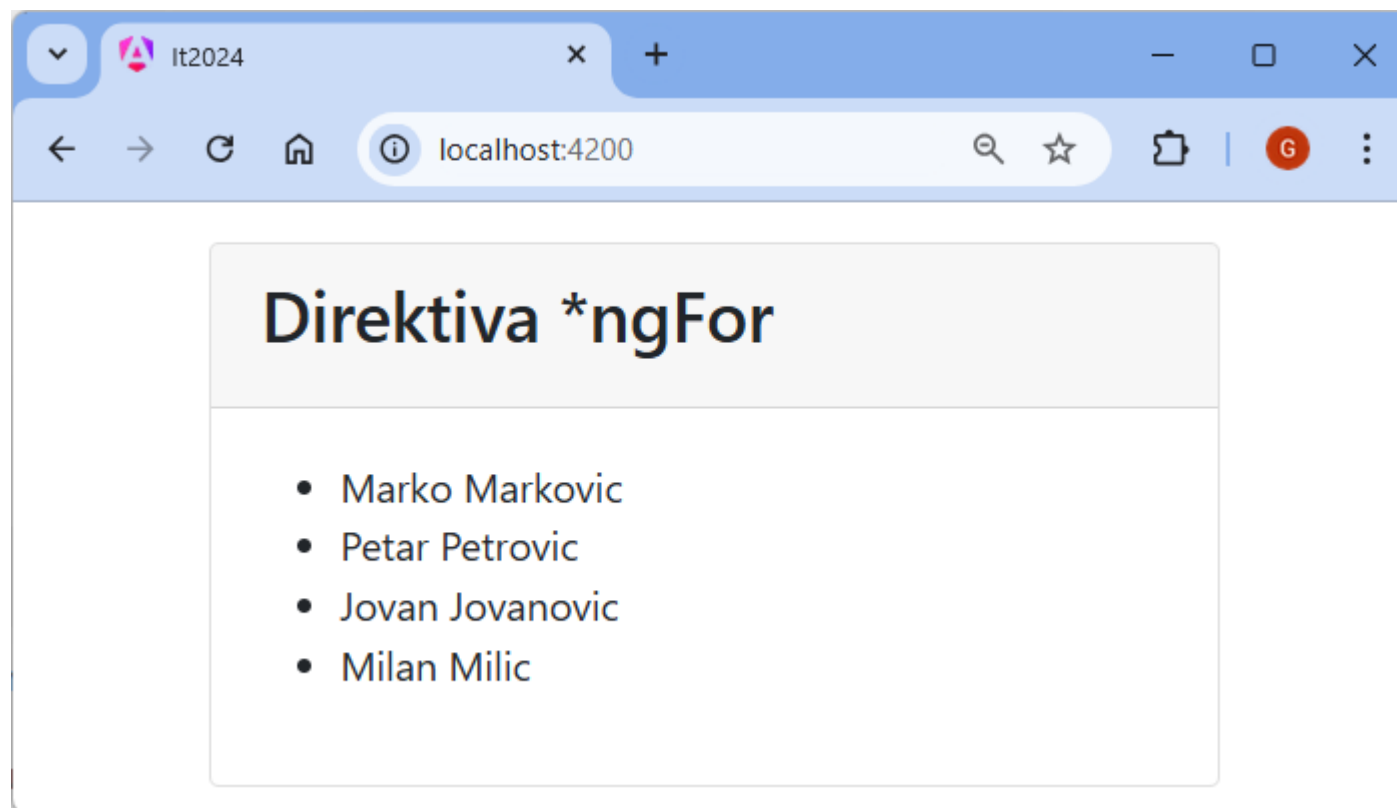
```
export class AppComponent
{

  osobe: Osoba[] = [
    new Osoba(1, 'Marko', 'Markovic'),
    new Osoba(2, 'Petar', 'Petrovic'),
    new Osoba(3, 'Jovan', 'Jovanovic'),
    new Osoba(4, 'Milan', 'Milic')
  ];
}
```


Direktiva *ngFor - primer

```
<div class="container mt-3">
  <div class="card">
    <div class="card-header">
      <h3>Direktiva *ngFor</h3>
    </div>
    <div class="card-body">
      <ul>
        <li *ngFor="let osoba of osobe">
          {{osoba.ime}} {{osoba.prezime}}
        </li>
      </ul>
    </div>
  </div>
</div>
```

Direktiva *ngFor



Lokalna varijabla u *ngFor

```
<div class="card">
  <div class="card-header">
    <h3>Direktiva *ngFor</h3>
  </div>
  <div class="card-body">
    <ul>
      <li *ngFor="let osoba of osobe; let i = index">
        {{i+1}} {{osoba.ime}} {{osoba.prezime}}
      </li>
    </ul>
  </div>
</div>
```

Pitanje 1

U klasi komponente definisan je niz

stavke: string[] = ['stavka1', 'stavka2', 'stavka3', 'stavka4'];

Potrebno je kreirati listu sa bulitima korišćenjem podataka iz niza definisanim u klasi komponente. Koristićemo

- a. Direktivu ngIf
- b. Direktivu ngFor
- c. Direktivu ngSwitch

Odgovor: b

Pitanje 2

Definisano je polje klase komponente tipa boolean pod nazivom **prikazi**. Za uslovno prikazivanje div elementa ako je polje prikazi postavljeno na vrednost **true** koristimo kod:

- a. `<div *ngFor="prikazi">`
- b. `<div *ngIf="prikazi">`
- c. `<div *ng="prikazi === true">`

Odgovor: b