

Uvod u Angular

NIKOLA AJZENHAMER

PROGRAMIRANJE ZA VEB

MATEMATIČKI FAKULTET

2020/2021

Sadržaj

Ugrađene Angular direktive

Tok podataka kroz hijerarhijsku organizaciju komponenti

Referencne promenljive šablona

Životni tok komponenti

Ugrađene Angular direktive

Direktive

Instrukcije u okviru DOM stabla

Instrukcija – operacija nad DOM stablom koja menja prikaz stranice, stilove elemenata, ...

Vrste direktiva

1. Komponente

- Instrukcija: dodaj HTML elemente u DOM stablo na osnovu pridruženog šablona i stila
- Direktive sa pogledom

2. Strukturne direktive

- Instrukcija: dodaj/ukloni HTML elemente u/iz DOM stablo/a
- Strukturne direktive ne prati HTML šablon

3. Atributske direktive

- Instrukcija: promeni izgled, ponašanje itd. HTML elementa, komponente ili druge direktive

Direktive

Često se na direktive referiše na dva načina

Velika kamilja notacija

- Klasa koja implementira instrukciju direktive
- Primer: `NgIf`

Mala kamilja notacija

- Ime atributa direktive; selektor
- Primer: `ngIf`

Strukturne direktive

NgIf

- Uklanjanje ili rekreiranje dela DOM stabla na osnovu vrednosti Bulovog izraza
- Ponašanje odgovara `if` klauzi u višim programskih jezicima

NgForOf

- Prikazuje kolekciju DOM podstabala na osnovu vrednosti niza
- Ponašanje odgovara `for` ciklusu u višim programskim jezicima

NgSwitch, NgSwitchCase, NgSwitchDefault

- Uklanjanje ili rekreiranje dela DOM stabla na osnovu vrednosti izraza
- Ponašanje odgovara `switch` klauzi u višim programskim jezicima

Zvezdasti prefiks

Strukturne direktive se najčešće primenjuju u svojoj skraćenoj sintaksičkoj formi

```
<div *ngIf="isVisible">I am visible!</div>
```

U svojoj punoj sintaksičkoj varijanti, strukturne direktive predstavljaju atribut nad HTML elementima u šablonima

```
<ng-template [ngIf]="isVisible">
  <div>I am visible!</div>
</ng-template>
```

Primenom zvezdastog prefiksa, Angular:

- Izmešta direktivu `*ngIf` iz `div` elementa u roditeljski element `ng-template` u kojem se ona koristi kao vezivanje atributa
- Ostatak HTML elementa smešta kao HTML sadržaj elementa `ng-template` bez upotrebe direktive

Atributske direktive

NgClass

- Postavljanje ili uklanjanje klasa na osnovu vrednosti objekta čiji su ključevi nazivi klasa, a vrednosti Boleve vrednosti

NgStyle

- Postavljanje ili uklanjanje CSS svojstva na osnovu vrednosti objekta čiji su ključevi nazivi CSS svojstva, a vrednosti Boleve vrednosti

Tok podataka
kroz hijerarhijsku
organizaciju komponenti

Prirodni tok podataka

U okviru jedne komponente

- Podaci teku između modela i pogleda

Između dve komponente

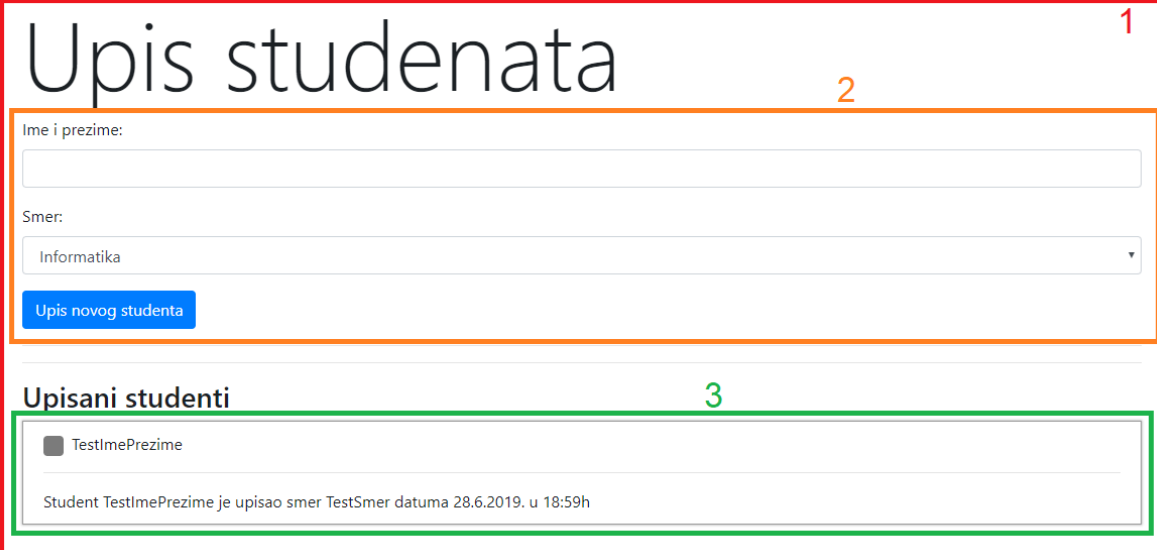
- Podaci teku između komponenti koje su u relaciji „je roditelj od“

Primer aplikacije na slike

- K1** „je roditelj od“ **K2**
- K1** „je roditelj od“ **K3**

Između **K2** i **K3** ne postoji prirodni tok podataka jer nisu u relaciji „je roditelj od“

- Postoje drugi mehanizmi za razmenu podataka (servisi)



The screenshot shows a web application interface for student registration. It is divided into three main sections, each highlighted with a colored border and a number:

- 1** (Red border): The title "Upis studenata" is located in the top right corner of the page.
- 2** (Orange border): This section contains the registration form with fields for "Ime i prezime:" (a text input) and "Smer:" (a dropdown menu currently showing "Informatika"). Below these fields is a blue button labeled "Upis novog studenta".
- 3** (Green border): This section is titled "Upisani studenti" and displays a list of registered students. The first entry shows a student icon, the name "TestImePrezime", and a log message: "Student TestImePrezime je upisao smer TestSmer datuma 28.6.2019. u 18:59h".

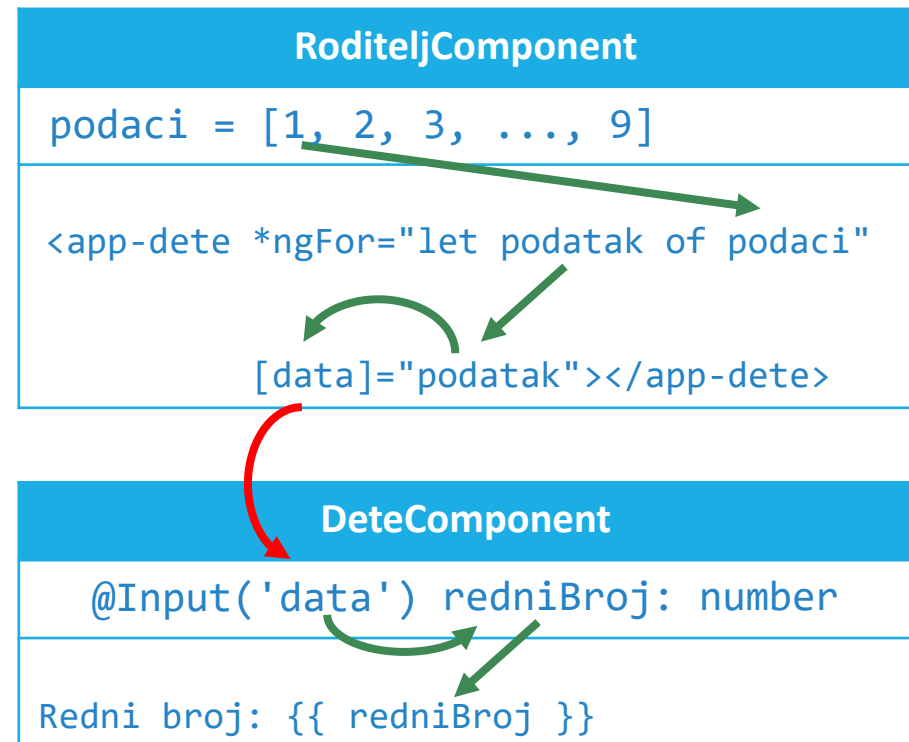
Ulazni tok

Mehanizam slanja podataka iz roditeljske komponente ka dete komponenti

- Slanje podataka **niz** hijerarhiju

Procedura za kreiranje ulaznog toka

- U dete komponenti kreiramo svojstvo odgovarajućeg tipa
- Svojstvo dekorišemo dekoratorom `@Input`
 - Opcioni argument je naziv atributa koji će roditelj videti
 - Ako se ne navede, podrazumeva se naziv svojstva
- U roditeljskoj komponenti koristimo sintaksu vezivanja **atributa** nad dete komponentom



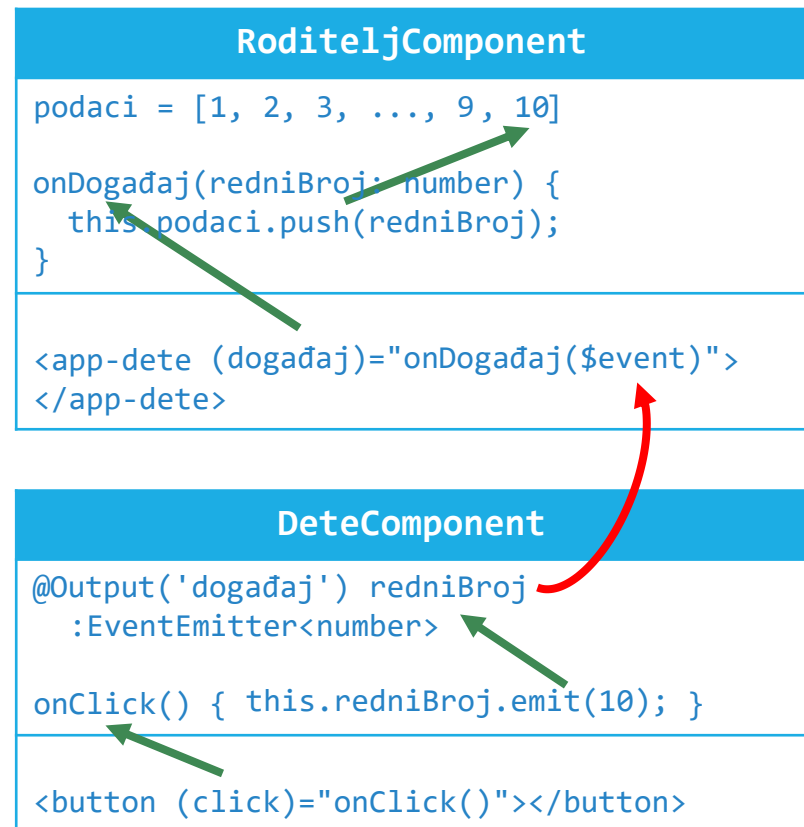
Izlazni tok

Mehanizam slanja podataka iz dete komponente ka roditeljskoj komponenti

- Slanje podataka **uz** hijerarhiju

Procedura za kreiranje izlaznog toka

- U dete komponenti kreiramo svojstvo tipa `EventEmitter<T>`
 - `T` je konkretan tip podatka koji se šalje
- Svojstvo dekorišemo dekoratorom `@Output`
 - Opcioni argument je naziv događaja koji će roditelj videti
 - Ako se ne navede, podrazumeva se naziv svojstva
- Podatak se šalje pozivom metoda `emit()` nad dekorisanim svojstvom u dete komponenti
- U roditeljskoj komponenti koristimo sintaksu vezivanja **događaja** nad dete komponentom
 - Opcioni argument `$event` predstavlja podatak koji je poslat



Referencne promenljive šablona

Referencne promenljive šablona

Uvode se navođenjem sintaksnog konstrukta `#identifikator`

Referenciraju se u šablonu navođenjem identifikatora (bez karaktera `#`)

Upotreba:

- U `NgIf` direktivi kao vrednost `else` klauze
- Anotiranje elemenata u pogledu radi prosleđivanja njihove reference u modelu
- Inicijalizacija promenljivih u modelu koje se odnose na neke HTML elemente iz šablona
 - Dekorator `@ViewChild`

Dekorator `@ViewChild`

Dekorator `@ViewChild` dekoriše svojstvo modela komponente čiji je tip `ElementRef`

`ElementRef` je tip podataka koji sadrži svojstvo `nativeElement` koje sadrži referencu na element iz pogleda te komponente

Argumenti dekoratora:

1. Upit kojim se dohvata prvi element
 - Naziv referencne promenljive šablona
 - Tip komponente
2. Objekat sa dodatnim opcijama
 - Opcija `static` tipa `boolean` definiše da li se element dohvata statički (pre ciklusa detekcije promena)
 - Navođenjem vrednosti `true`, svojstvo se inicijalizuje pred poziv metoda `ngOnInit()`
 - Navođenjem vrednosti `false`, svojstvo se inicijalizuje pred poziv metoda `ngAfterViewInit()`

Životni tok komponenti

Životni tok komponenti

Svaka komponenta ima svoj životni tok koji Angular održava

U toku ovog životnog veka, Angular:

- Kreira komponentu
- Ažurira prikaz u pogledu
- Upravlja izmenama i reagovanjima na događaje
- Uništava ih pre uklanjanja iz DOM stabla
- ...

Osluškivanje životnog toka

Moguće je implementirati metode koji će biti pozvani u odgovarajućim trenucima u toku životnog toka

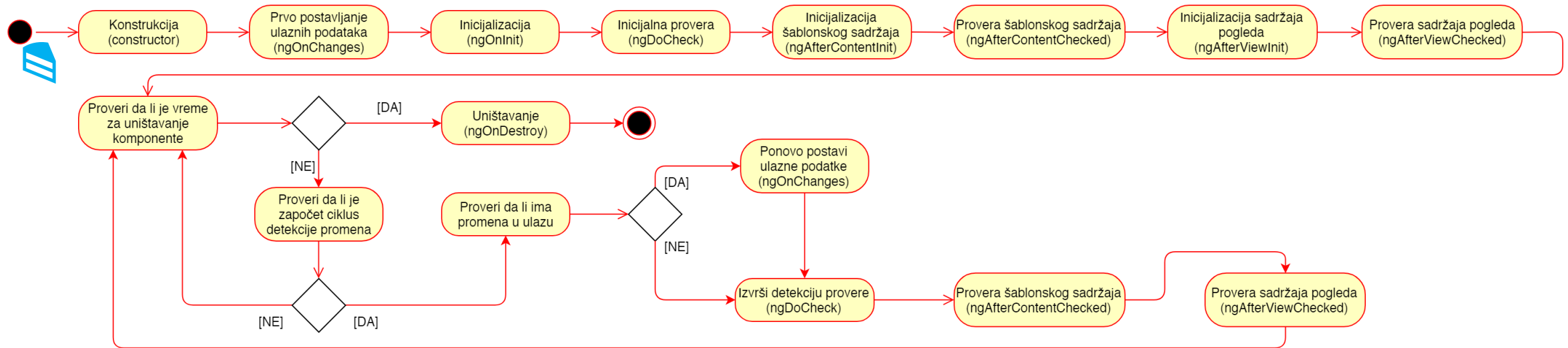
Potrebno je da komponenta

- Implementira odgovarajuće interfejse
- Implementira odgovarajući metod iz datog interfejsa

Primeri interfejsa i metoda

Interfejs	Metod	Trenutak u životnom toku
<code>OnInit</code>	<code>ngOnInit</code>	Inicijalizuje se nakon prvog prikaza i postavljaju se podaci iz ulaznog toka. Poziva se jedanput, nakon prvog poziva metoda <code>ngOnChanges()</code> .
<code>OnChanges</code>	<code>ngOnChanges</code>	Prilikom (ponovnog) postavljanja vrednosti za ulazne podatke.
<code>OnDestroy</code>	<code>ngOnDestroy</code>	Poziva se tik pred uništenje komponente. Korisno za izbegavanje curenje memorije.
<code>DoCheck</code>	<code>ngDoCheck</code>	Za trenutke koje Angular ne može da detektuje. Poziva se u svakom ciklusu detekcija promena, neposredno nakon <code>ngOnChanges()</code> i <code>ngOnInit()</code> .

Dijagram životnog toka komponente



Hvala na pažnji!

NIKOLA AJZENHAMER

PROGRAMIRANJE ZA VEB

MATEMATIČKI FAKULTET

2020/2021