M&I SYSTEMS GROUP

VOJVODINA ICT CLUSTER

Angular

GO4Code

Marko Cvijanović
Mihailo Bubnjević

Anatomija komponente

- Komponenta je TS klasa dekorisana dekoratorom **@Component**
- **selector** - naziv elements koji će se koristiti za dodavanje u HTML
- **templateUrl** - url HTML templejta
- **styleUrls** - stilovi koji se primenjuju samo na ovu komponentu

- Kada je komponenta kreirana potrebno je registrovati je u aplikaciji
- Registrovanje komponente se obavlja dodavanjem komponente u deklaracije ngModule-a

Moduli

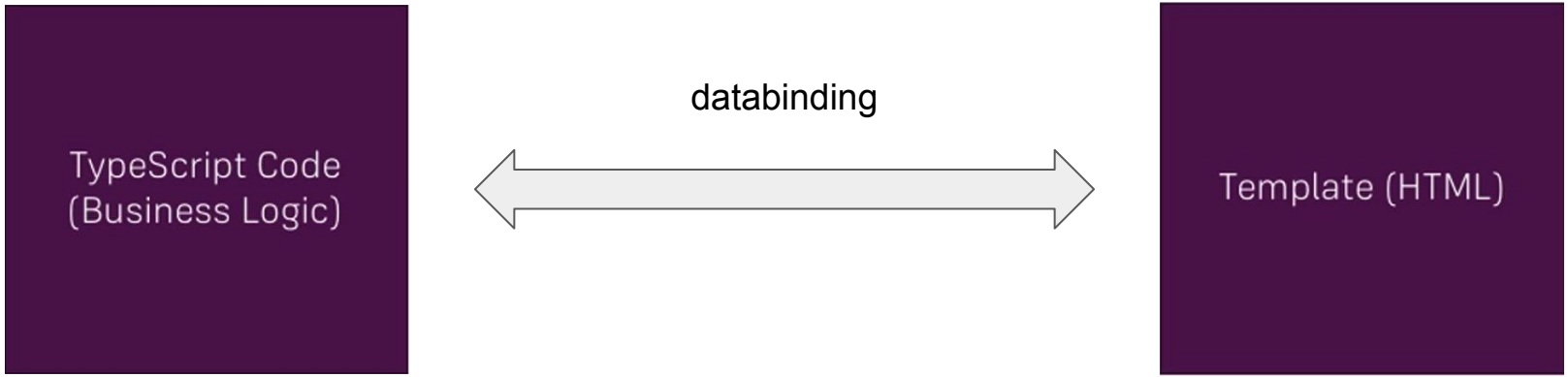
- Moduli su klase anotirane sa @NgModule
- Svaka aplikacija ima makar jedan ngModule, to je korenski modul (AppModule)
- Problem: vremenom AppModule postaje nepregledan u velikim aplikacijama
- Rešenje: uvode se featur moduli
  - ngModule koji je posvećen jednom feature-u aplikacije
- Komponente koje spadaju u isti feature se registruju u njegov feature modul
- Feature module je veoma sličan korenskom sa par razlika
  - Korenski modul se bootuje pri pokretanju aplikacije, dok se feature module importuje
  - Feature module može da sakrije deklaracije od drugih ngModula

NgModule

- 3 bitna dela @NgModule-a
  - **declarations** - komponente koje su definisane u modulu
  - **exports** - komponente koje želimo da podelimo sa drugim modulima
  - **imports** - komponente koje želimo da importujemo tj. dodamo u modul

Databinding

- Način na koji typescript kod komunicira sa html-om
- Mehanizmi koje koristimo za databinding u angularu:
  - String interpolation
  - Property binding
  - Event binding
  - Two-way binding



String interpolation

- Za polje data u typescript fajlu, koristimo {{ data }} sintaksu kako bismo pristupili vrednosti unutar html-a
- Data može predstavljati bilo koji typescript izraz koji možemo evaluirati na string (implicitne konverzije takođe)
- Data ne može predstavljati blok koda (if, else, for)
- Možemo koristiti metode čiji povratni tip možemo evaluirati na string
- Ternarni operator je takođe dozvoljen

Property binding

- Za property na html elementu možemo definisati property binding tako što ćemo property staviti u uglaste zagrade i dodeliti mu vrednost unutar navodnika
- [property]="value" -> value predstavlja podatak unutar typescript fajla
- Svaki property ima očekivani tip vrednosti

Event binding

- Koristimo kada želimo da pozovemo neku funkciju nakon nekog događaja iz html-a
- (event) = "onEvent()"
- Umesto funkcije možemo direktno pisati kod unutar navodnika
- Event može proslediti neke parametre u poziv funkcije, gde tip i vrednost tih parametara zavisi od konkretnog event-a

Two-way databinding

- Sadrži u sebi mehanizme za property binding kao i event binding
- Sintaksa je kombinacija sintaksi za property i event binding
- [(ngModel)] = "value"

Direktive

- Instrukcije unutar DOM stabla
- Komponente su zapravo direktive sa templejtom
- Kada kažemo direktive u glavnom mislimo na direktive bez templejta
- Direktive koristimo kao property-je nad nekim html elementom kako bismo mu dali instrukciju

Direktive

- Ugrađene direktive:
  - ngIf
  - ngFor
  - ngSwitch
  - ngStyle
  - ngClass

Direktive

- **ngIf**
- Ukoliko je zadovoljen zadati uslov element nad kojim je postavljena ova direktiva biće ugrađen u DOM stablo
- *ngIf="<neki truthy ili falsy izraz ili vrednost>"
- Ako je falsy element se uopšte neće ni prikazati ni napraviti u Dom stablu
- Ako je truthy prikazaće se i biće dostupan

Direktive

- **ngIf**
- Možemo da napišemo i else blok koristeći
  - <ng-template> ugrađeni angular element koji se ne ispisuje direktno u DOM
  - #<imeReference> - lokalna referenca

```
<p *ngIf="condition; else elseBlock">Write something</p>
<ng-template #elseBlock>
    <p>Write something else</p>
</ng-template>
```

Direktive

- **ngStyle**
- Dopušta nam da dinamički menjamo izgled elementa
- [ngStyle]="{<camelCaseCssProperty> : vrednost}"

```
<p [ngStyle]="{color: 'red'}"></p>
```

Direktive

- **ngClass**
- Dopušta nam da dinamicki dodajemo i oduzimamo css klase nekom elementu na osnovu nekog uslova
- Ako je uslov true klasa će biti dodata
- Ako je uslov false klasa će biti oduzeta
- [ngClass]="{<imeKlase> : <condition>}"

```
<p [ngClass]="{className : condition}"></p>
```

Direktive

- **ngFor**
- Direktiva za iteraciju nad grupom elemenata
- definišemo lokalnu varijablu sa **let** koju možemo da koristimo unutar elementa
- *ngFor="let <item> of <items>"

```
<p *ngFor="let item of items">{{item}}</p>
```

Direktive

- **ngSwitch**
- Koristi se kao switch case u drugim programskim jezicima
- [ngSwitch]=”switch_expression”
- *ngSwitchCase=”match_expression_1”
  *ngSwitchCase=”match_expression_2”..
- ngSwitchDefault

Direktive

- Možemo da definišemo svoje direktive
- koristimo CLI komandu **ng generate directive <imeDirektive>**
- Nova direktiva mora biti dodata unutar modula
- Zahteva referencu na element na koji se primenjuje unutar svog konstruktora

```
constructor(private elementRef: ElementRef) {
```

- Unutar konstruktora možemo definisati ponašanje tog elementa npr. promeniti mu stil