

Struktura Angular projekta

1. Struktura Angular projekta

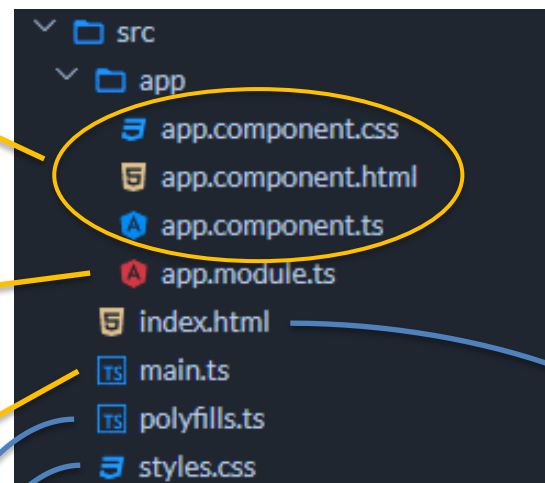
Angular aplikacije se sastoje od mnoštva komponenti, a komponente koje grade neku logičku celinu grupišu se u module. Kao što već znamo, svaka Angular aplikacija ima bar jedan korenski modul, koji sadrži korensku komponentu. Ostale komponente se ugnježdavaju unutar korenske komponente.

- ! Zašto je bitno upoznati se sa strukturom Angular projekta?
 - ! Poznavanje arhitekture Angular projekta doprinosi efikasnijem i efektivnijem radu.

app.component.ts je komponenta koju ugrađujemo u `index.html`; komponenta ima svoj šablon (`app.component.html`) i stilove (`app.component.css`)

app.module.ts je korenski modul aplikacije (objedinjuje komponente i servise)

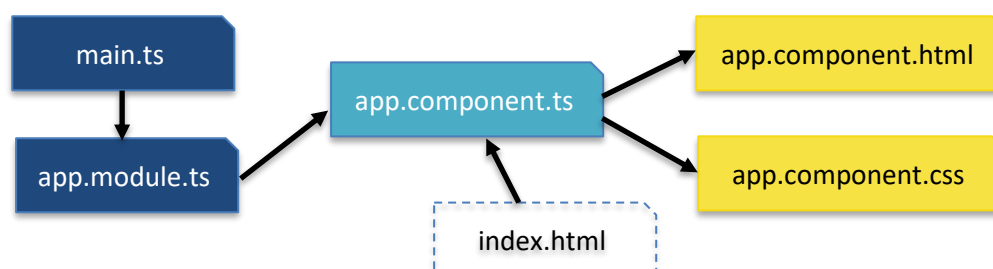
main.ts je odgovoran za pokretanje aplikacije (vrši potrebna učitavanja i startuje aplikaciju)



index.html je osnovni HTML fajl od kog počinje formiranje web stranice

styles.css definiše globalne stilove na nivou cele aplikacije

polyfills.ts sadrži pomoćne skripte za postizanje kompatibilnosti aplikacije sa starijim web browseri-ima



1.1 Angular komponente

Komponenta upravlja delom prikaza koji zovemo pogled (*view*). Komponenta je definisana **dekoratorom** i **klasom**:

```
@Component({
  selector: 'super-cool-list',
  templateUrl: './super-cool-list.component.html',
  styleUrls: ['./super-cool-list.component.css']
})
export class SuperCoolListComponent {
  /* ... */
}
```

Klasa implementira logiku koja se nalazi u pozadini pogleda. U primeru iznad, `SuperCoolListComponent` je obična klasa i ne predstavlja komponentu sve dok se ne obeleži `@Component` dekoratorom.

Dekorator `@Component` identifikuje klasu koja se nalazi ispod dekoratora kao komponentu i specificira meta podatke klase:

- * `selector` definiše naziv elementa putem kog će se komponenta ugraditi u HTML ostatka aplikacije
- * `templateUrl` je lokacija HTML fajla koji predstavlja šablon komponente
- * `styleUrls` je lista lokacija CSS fajlova sa stilovima koje koristi šablon komponente



Koje su prednosti razvoja aplikacije putem komponenti?

1. Hijerarhija komponenti: Između komponenti se može uspostaviti hijerarhijski odnos tako što se unutar jedne komponente koristi druga komponenta (selektor jedne komponente se navede u šablonu druge komponente).
2. Lakši razvoj: Raščlanjavanje prikaza podataka olakšava razvoj što posebno dolazi do izražaja kod složenijih aplikacija.
3. Lakši timski rad: Podela posla unutar tima može se vršiti po komponentama i time se olakšava timski rad na razvoju aplikacije.

1.2 Angular šabloni



Angular komponente koriste šablone za formatiranje prikaza i za određivanje načina interakcije sa korisnikom. Jezik za pisanje šablona je zasnovan na HTML-u.

Komponente i šabloni su povezani pomoću `@Component` dekoratora. Postoje dva načina za definisanje šablona:

1. svojstvom `template` čija vrednost sadrži šablon
2. svojstvom `templateUrl` čija vrednost je putanja do spoljašnjeg fajla u kom je šablon

Prvi način (pisanje šablona u okviru dekoratora) može učiniti komponentu nepreglednom, pa se češće koristi **drugi način**.

```
@Component({
  selector: 'super-cool-list',
  template: '<p>Super Cool List!</p>',
  styleUrls: [ './ super-cool-list.component.css' ]
})
```

```
@Component({
  selector: 'super-cool-list',
  templateUrl: './super-cool-list.html',
  styleUrls: [ './ super-cool-list.component.css' ]
})
```

1.2.1 Sintaksa Angular šablona

U osnovi Angular šablona je HTML. Komponente u svojim šablonima definišu odgovarajuće segmente web stranice tako da u celini čine validnu HTML stranicu. Neki HTML elementi, kao što su `<html>` i `<body>`, nemaju ulogu u šablonima. Element `<script>` je zabranjen u šablonima kako bi se uklonili rizici od napada (script injection attacks). Sve ostalo ostaje poprilično isto kao i kod običnih HTML fajlova.

HTML sintaksu je moguće proširiti komponentama i direktivama koje Angular pruža. Jedan način da se HTML šablon obogati jeste korišćenje **interpolacije**. Interpolacija se koristi kako bi se kroz šablon ugradila vrednost nekog podatka u web stranicu. Interpolacija se postiže navođenjem izraza unutar duplih vitičastih zagrada `{{ izraz }}`.

1.3 Dodatni moduli

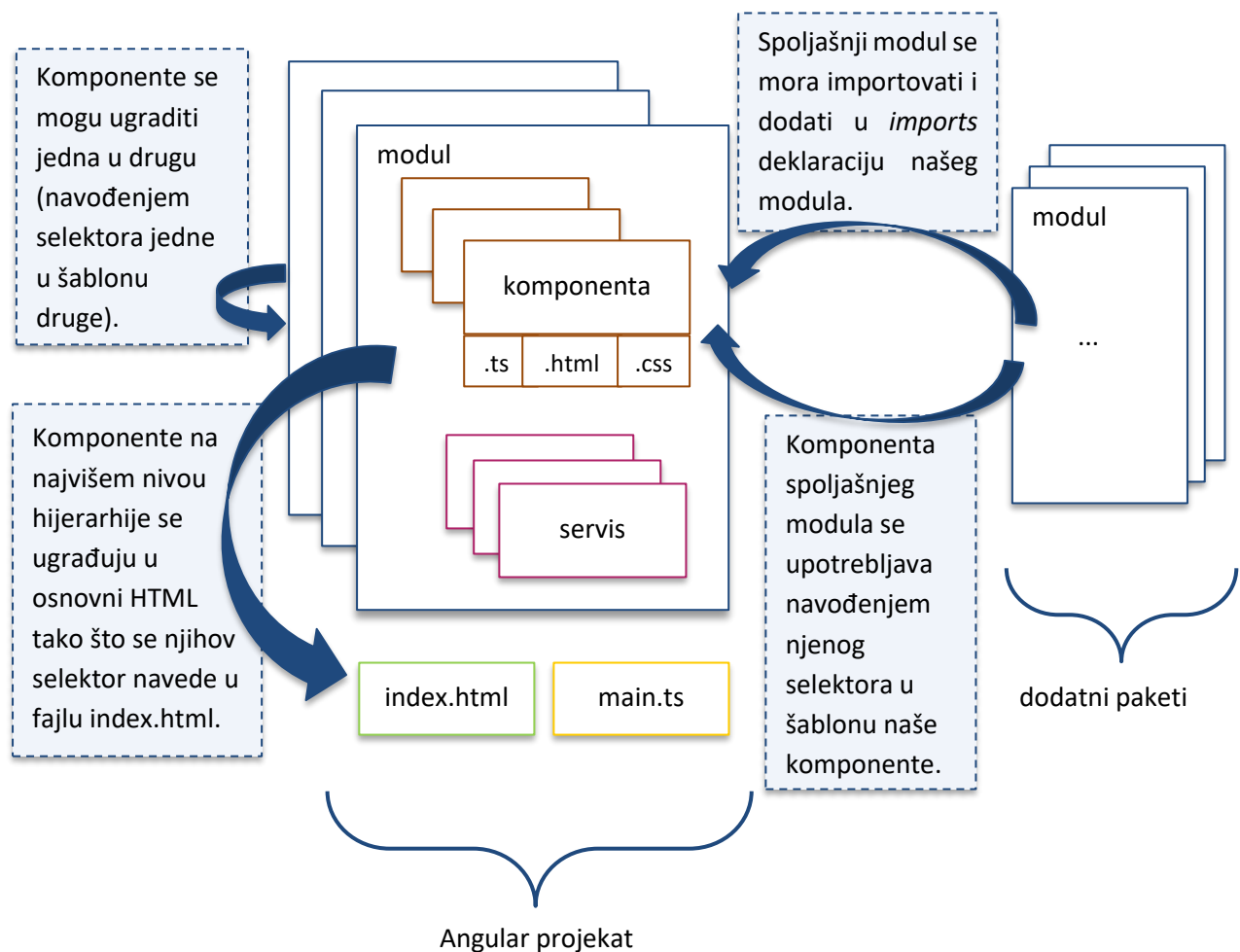
Pored modula koji pripadaju Angular projektu, moguće je ugrađivanje dodatnih mogućnosti korišćenjem spoljašnjih modula. Ti moduli se nazivaju *dependencies* tj. zavisnosti, jer njihovim korišćenjem projekat zavisi od tih modula.

Da bi uključili neki modul u Angular projekat potrebno je znati naziv paketa putem kojeg se taj modul distribuira. Za preuzimanje paketa, u *dependencies* delu projekta je potrebno upisati naziv paketa, nakon čega će početi njegovo preuzimanje. Ukoliko moduli zavise od nekih drugi modula, oni će biti automatski preuzeti.

! Pošto je razvoj modula kontinuiran proces, verzije paketa mogu biti od velikog značaja za međusobnu kompatibilnost više modula.

1.3.1 npm – Node Package Manager

Preuzimanje paketa koji su u Angular projektu dodati kao zavisnosti (*dependencies*) vrši alat **npm** (Node Package Manager). Na sajtu **npmjs.com** dostupan je pregled raspoloživih paketa koje je moguće koristiti kao zavisnosti. Postoji više od milion paketa (namenjeni Angular aplikacijama, ali i mnogim drugim koje koriste JavaScript jezik), te je moguće izvršiti pretragu i rangiranje kako bi se lakše pronašli željeni paketi.



2. Angular Material

Material Design je unificiran jezik za dizajniranje aplikacija koji je prilagodljiv različitim uređajima i različitim veličinama ekrana. **Angular Material** je kolekcija komponenti saglasnih Material Design specifikaciji i namenjenih korišćenju u okviru Angular projekata. Angular Material se sastoji od više modula u kojima se nalaze Material Design komponente.

Paket u kojem su objedinjeni ovi moduli se zove **@angular/material**. Upisivanjem ovog paketa u *dependencies* odeljak Angular projekta biće preuzet ceo paket. Za pravilan prikaz komponenti, na početku je potrebno odabrati temu kojom će komponente biti stilizovane – u fajl *styles.css* se ugradi tema @import deklaracijom: `@import '@angular/material/prebuilt-themes/deeppurple-amber.css';`

2.1 Primer za Material Card

```
import { MatCardModule } from '@angular/material/card';
...
@NgModule({
  imports: [..., MatCardModule ],
  ...
})
export class AppModule { }
```

Za korišćenje neke od komponenti u šablonu, potrebno je:

- importovati modul
- dopisati naziv importovanog modula u *imports* odeljak

Zatim, u šablon komponente ugraditi direktive koje koristi ova komponenta.

Zadaci

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

1. Kreirati komponentu za prikaz podataka o omiljenim pesmama (title1, artist1, albumCover1; title2, artist2, albumCover2,...) uz korišćenje Angular Material biblioteke.
 - * Dodati link do pesme na YouTube-u
2. Kreirati komponentu za prikaz podataka o omiljenim filmovima (title1, genre1, poster1; title2, genre2, poster2,...) uz korišćenje Angular Material biblioteke.
 - * Dodati link do filma na IMDb-u

Domaći

1. Kreirati komponentu za prikaz podataka o omiljenim knjigama (title1, author1, genre1, bookCover1, description1; title2, author2, genre2, bookCover2, description2,...) uz korišćenje Angular Material biblioteke.
 - * Dodati link do knjige na Goodreads-u

Literatura

1. <https://angular.io/>
2. <https://www.npmjs.com/>
3. <https://material.angular.io/>
4. Materijali sa predavanja