

Rutiranje

Konfiguracija projekta

Servisi: simulacija pozadinske

logike

Angular Material uvod

Dr. Mlađan Jovanović
mjovanovic@singidunum.ac.rs

Sadržaj

- ◆ Rutiranje
- ◆ Kreiranje i konfiguracija projekta
- ◆ Servisi
 - Slučajevi korišćenja kao simulacija pozadinske logike
- ◆ Angular Material komponente korisničkog interfejsa:
- ◆ Instalacija
- ◆ Kolekcija komponenti
- ◆ Teme
- ◆ Korišćenje (dugme, simboli)

Model navigacije klijentskih Web aplikacija

◆ Klasičan model navigacije:

- ◆ Korisnik unosi URL adresu i pretraživač otvara datu stranicu
- ◆ Birajući linkove na stranici, pretraživač upućuje korisnika na nove stranice
- ◆ Forward i back dugmad se mogu koristiti za kretanje kroz istorijat posećenih stranica

◆ Angular model navigacije:

- ◆ Analogno modelu iznad, sa tim što se navigacija definiše do komponenti
 - ◆ Ono što se prikazuje je templejt komponente (html)
 - ◆ Istorijat se čuva na identičan način (back/forward)

Rutiranje

- ◆ Angular mehanizam koji omogućava definisanje putanja do komponenti aplikacije i njihovo prikazivanje
- ◆ Putanja se definiše do komponente, a ne stranice
- ◆ Različite komponente se prikazuju u okviru jedne stranice
- ◆ SPA (Single Page Application)
- ◆ Nema ponovnog učitavanja čitave stranice, već samo dela (taga) u okviru kojeg se prikazuje komponenta

Definisanje ruta

- 1. Učitavanje potrebnih modula**
- 2. Definisanje ruta**
- 3. Definisanje mesta prikazivanja komponenti
na koje ukazuju rute**
- 4. Definisanje linkova ka rutama**

R1 - Učitavanje potrebnih modula

Angular Router je opcionalni deo Angular-a čije su funkcionalnosti implementirane u biblioteci [@angular/router](#)

Na nivou Angular aplikacije se kreira jedna instanca Router servisa (*singleton*)

RouterModule – ugrađeni Angular modul koji implementira mehanizam rutiranja

Routes – Struktura podataka koja definiše niz ruta.

Rute su predstavljene kao parovi

putanja do komponente : naziv komponente

```
import {Routes, RouterModule} from '@angular/router';
```

R2 – Definisanje ruta

```
const rute:Routes = [
  {path: '', component: WelcomeComponent},
  {path: 'signup', component: SignupComponent},
  {path: 'login', component: LoginComponent},
  {path: 'training', component: TrainingComponent}
]
@NgModule({
  imports: [RouterModule.forRoot(rute)],
  exports: [RouterModule]
})
```

R2 – Definisanje ruta

- ◆ Angular aplikacija ima jednu instancu Router-a
- ◆ Prilikom navođenja URL adrese u pretraživaču, Router traži rutu na osnovu path atributa kako bi identifikovao komponentu koju treba da prikaže
- ◆ Rute se definišu kao niz Routes objekata
 - const rute : Routes
 - Prazna putanja ('') predstavlja home stranicu aplikacije
- ◆ Routes objekat ima dva atributa
 - path – relativna putanja do komponente (bez leading slash '/')
 - componente – naziv komponente do koje vodi putanja
- ◆ Naredba u imports atributu modula konfiguriše Router za datu Angular aplikaciju
 - RouterModule.forRoot(rute)

R3 – Definisanje mesta prikazivanja komponente

router-outlet: tag koji označava mesto u okviru templejta korene komponenta (app.component.html) na kojem se prikazuje sadržaj komponente do koje vodi ruta
Placeholder za komponentu koja će biti prikazana kada se unese odgovarajuća URL adresa
Router na osnovu unete URL adrese pronađe komponentu i prikazuje kao sadržaj router-outlet taga

```
<router-outlet></router-outlet>
```

R4 – Definisanje linkova ka rutama

Elementi korisničkog interfejsa koji korisniku omogućavaju navigaciju

Mogu se navoditi kao linkovi (`<a>`) sa direktivom `routerLink` koja ukazuje na konkretnu rutu

Navodi se sa leading slash '/' simbolom

Može se korisiti i `href`

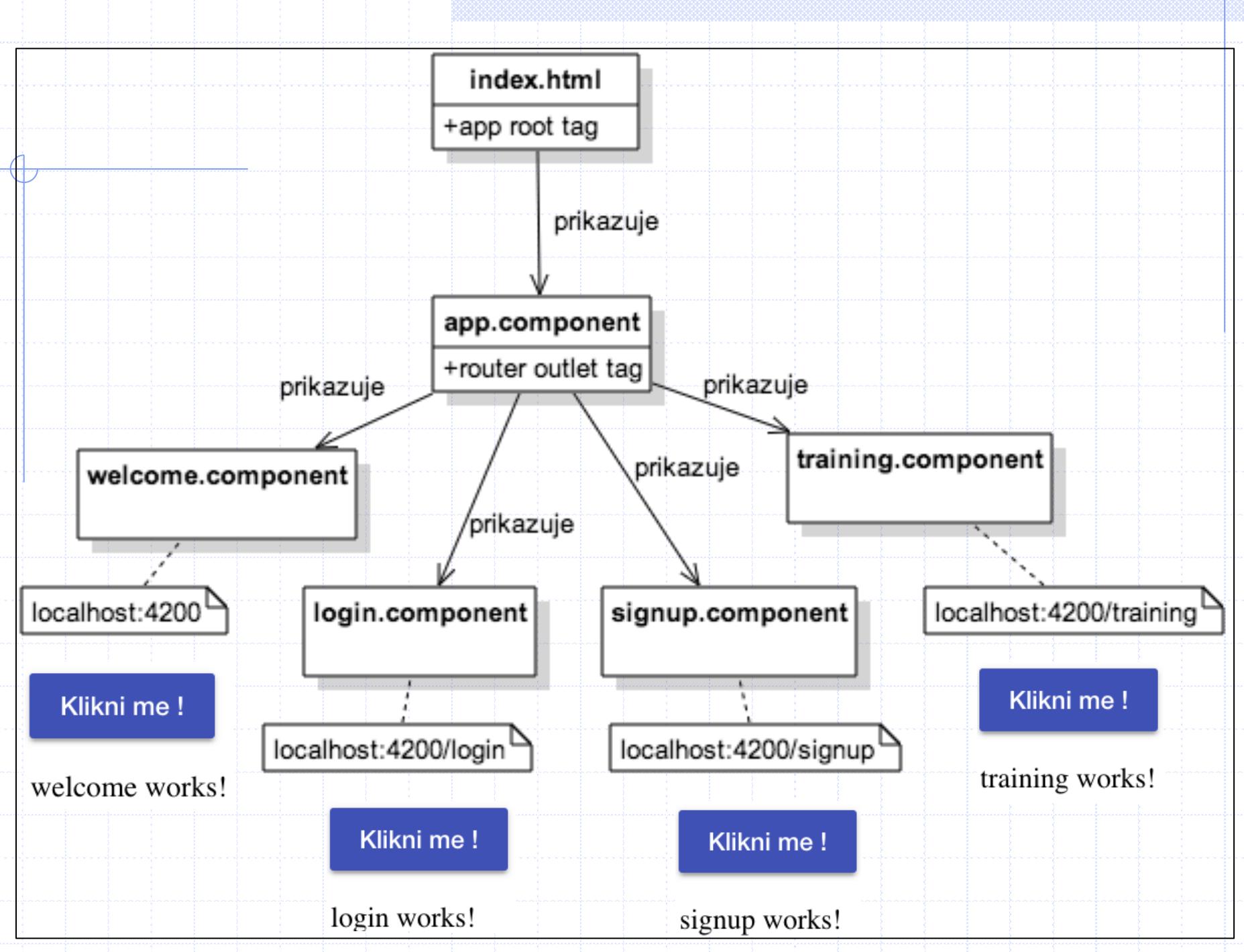
Vrednosti direktiva moraju odgovarati onima specificiranim u definiciji ruta

```
<a routerLink="/signup">Signup</a>
<a routerLink="/login">Login</a>
<a routerLink="/training">Training</a>
<a routerLink="/">Logout</a>
```

routing.module.ts

```
import { NgModule } from '../../../../../node_modules/@angular/core';
import { Routes, RouterModule } from '@angular/router';
import { WelcomeComponent } from './welcome/welcome.component';
import { SignupComponent } from './auth/signup/signup.component';
import { LoginComponent } from './auth/login/login.component';
import { TrainingComponent } from './training/training.component';

const route: Routes = [
{path : '', component: WelcomeComponent},
{path : 'signup', component: SignupComponent },
{path : 'login', component: LoginComponent},
{path : 'training', component: TrainingComponent}
]
@NgModule ({
    imports: [RouterModule.forRoot(route)],
    exports: [RouterModule]
})
export class RoutingModule {}
```



Rutiranje – termini i njihovo značenje

Koncept	Opis
RouterModule	Ugrađeni Angular module koji implementira funkcionalnosti rutiranja
Routes	Definiše niz ruta, gde svaka ruta definiše URL putanju do komponente (path-component)
RouterOutlet	HTML direktiva (<router-outlet>) koja definiše mesto prikazivanja templejta komponente Placeholder element
RouterLink	Atribut HTML taga koji sadržu putanju do rute (link ka ruti za korisnika)

Angular – osnovne instalacije

◆ Visual Studio Code:

<https://code.visualstudio.com/Download>

- Razvojno okruženje za Angular aplikacije

◆ Node.js: nodejs.org/download (preporucena verzija)

- Instalacija **Node Package Manager-a** (npm)

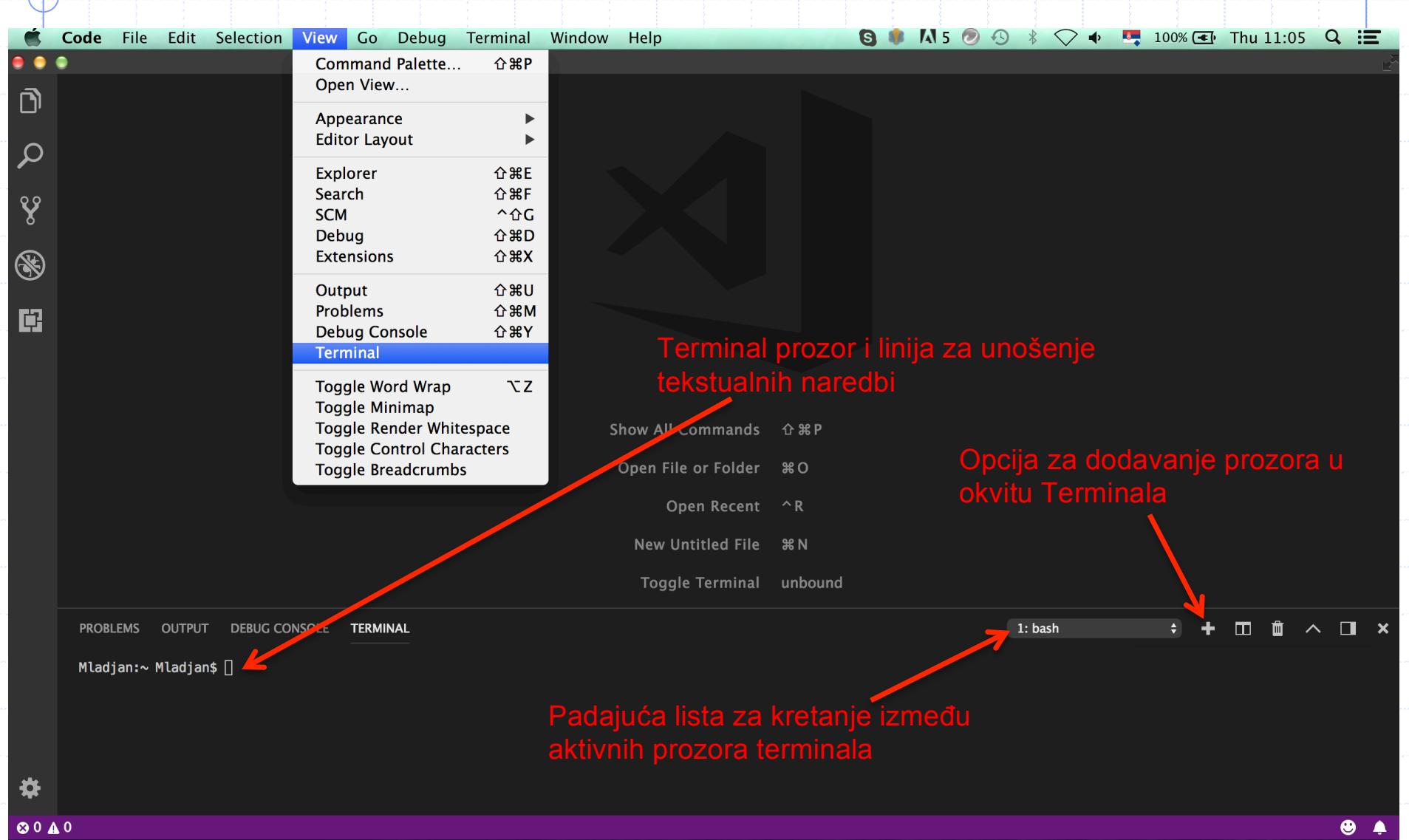
◆ Angular CLI: npm install -g @angular/cli

- Angular interfejs komandne linije (**Command Line Interface**)
- Programsko kreiranje projekta, komponenti, servisa, ...
- Pokretanje potrebnih instalacija

◆ Podrška za Angular Material komponente korisničkog interfejsa

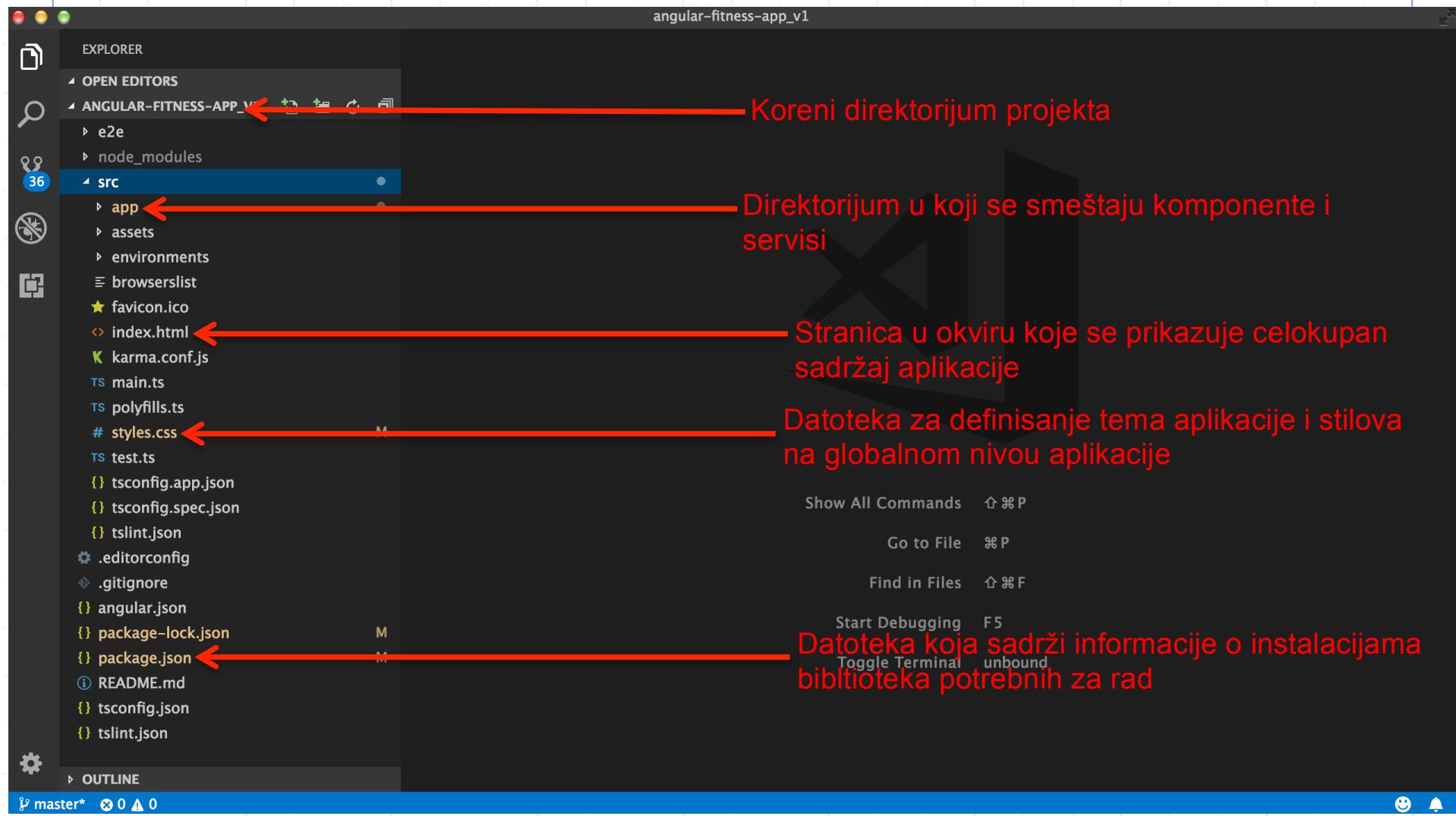
- npm install --save @angular/material @angular/cdk
@angular/animations

Razvojno okruženje



Kreiranje Angular projekta

`ng new ime_projekta`



Struktura Angular projekta

- ◆ **/src/**

- Kod Angular aplikacije

- ◆ **/node_modules/**

- npm paketi instalirani za projekat korišćenjem npm install naredbe

- ◆ **package.json**

- Zavisnosti (dependencies) kao verzije instaliranih biblioteka neophodne za prevođenje i izvršavanje programa

- ◆ **tsconfig.json**

- Konfiguracioni fajl za TypeScript kompajler

/src/ direktorijum Angular projekta

◆ **/app/**

- Elementi aplikacije (komponente, moduli i servisi)

◆ **/environments/**

- Promenjive okruženja (na primer, razvojno ili produpciono okruženje)

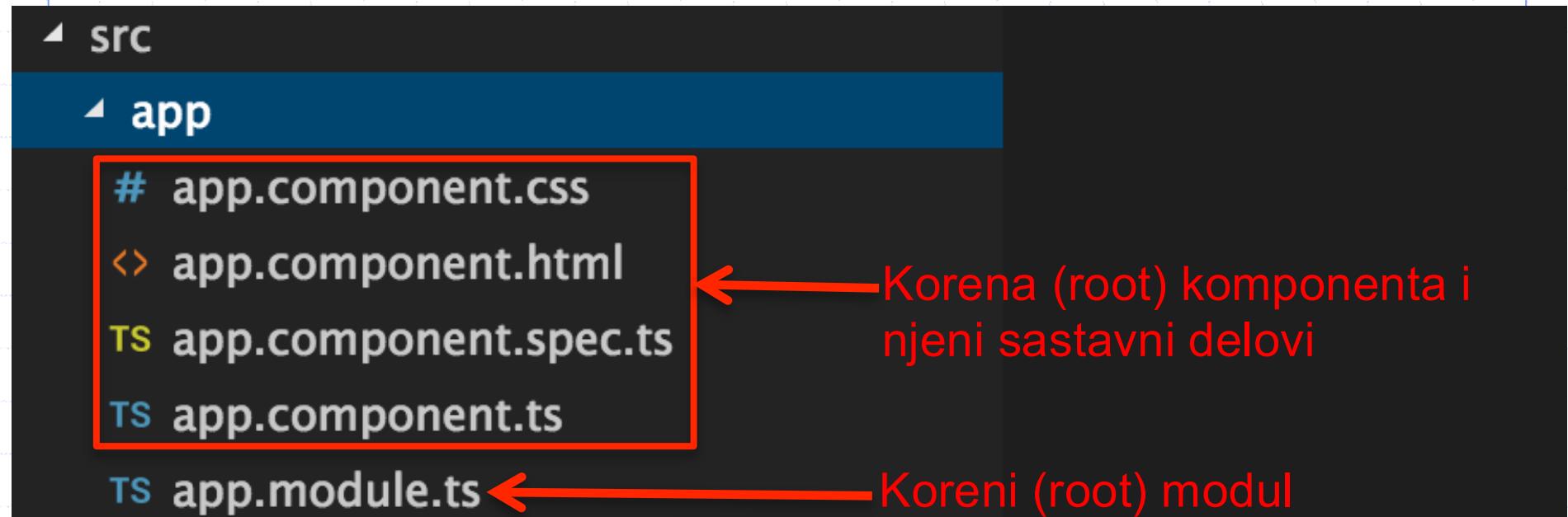
◆ **index.html**

- Home stranica Angular aplikacije
- SPA (Single Page Application), Single Page ≈ index.html

◆ **/assets/**

- Ovde se mogu smestiti objekti koje koristi aplikacija (slike, testni podaci i slično)

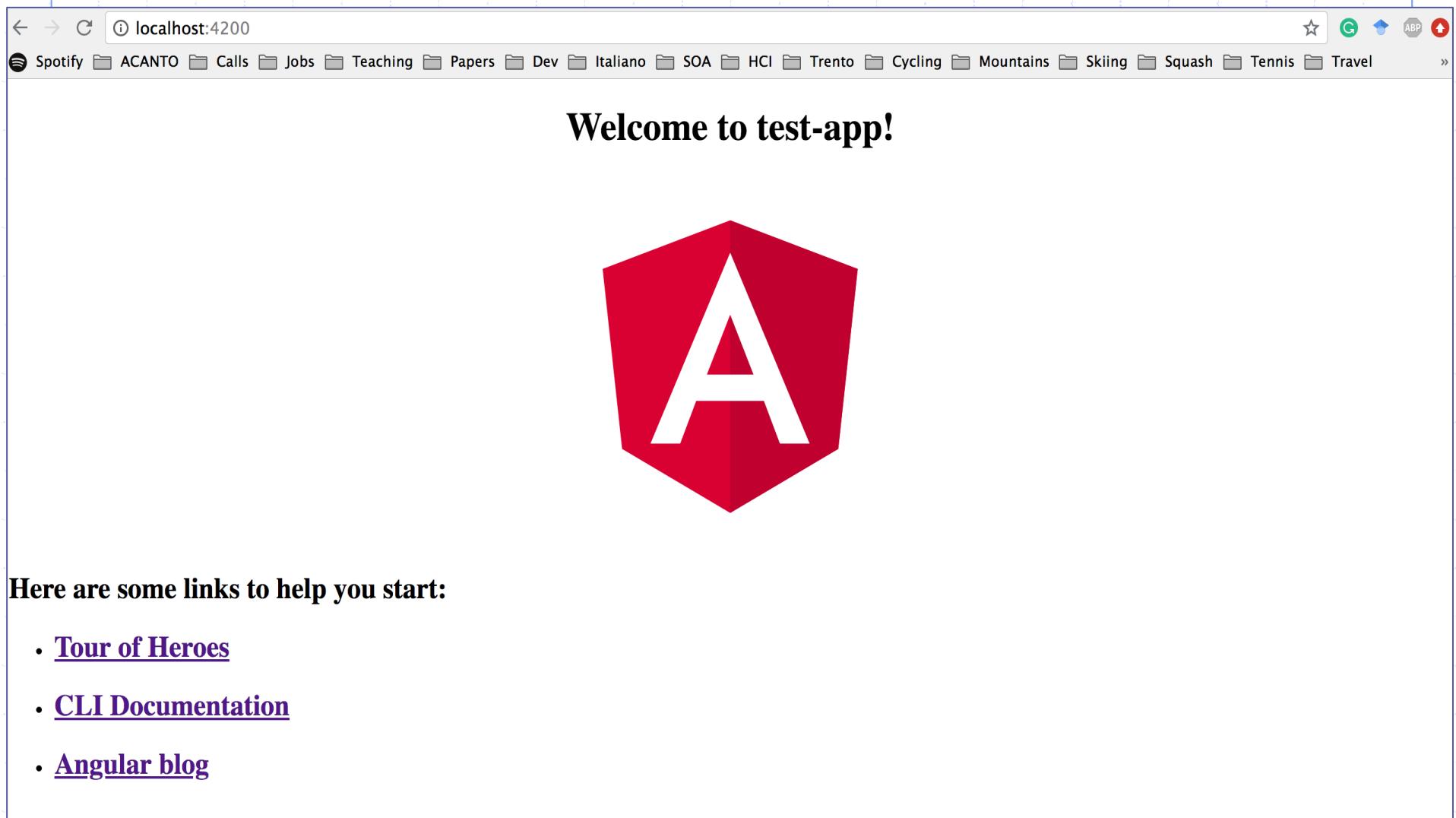
Korena komponenta i modul



Razvojni Web server

- ◆ Testiranje aplikacije iz razvojnog okruženja
- ◆ Terminal prozor CLI (interfejs komandne linije):
 - ng serve**
 - ng s**
- ◆ Testiranje aplikacije iz razvojnog okruženja
- ◆ Server automatski registruje i učitava promene prilikom svakog pamćenja promena (save)
- ◆ U nekim slučajevima se mora restartovati (ctrl+c)
 - Na primer, definisanje ruta
- ◆ Adresa razvojnog Web servera: localhost:4200

Razvojni server – nov projekat



A screenshot of a web browser window titled "localhost:4200". The address bar shows the URL. The browser interface includes standard navigation buttons (back, forward, search) and a toolbar with icons for Spotify, ACANTO, Calls, Jobs, Teaching, Papers, Dev, Italiano, SOA, HCI, Trento, Cycling, Mountains, Skiing, Squash, Tennis, and Travel. Below the toolbar, the main content area displays the text "Welcome to test-app!" in a large, bold, black font. In the center of the page is the large red Angular logo, which is a red hexagon containing a white stylized letter 'A'. At the bottom left of the page, there is a section of text that reads: "Here are some links to help you start:" followed by a bulleted list of three items: "Tour of Heroes", "CLI Documentation", and "Angular blog", each with a blue underline.

- [Tour of Heroes](#)
- [CLI Documentation](#)
- [Angular blog](#)

Programsko kreiranje komponenti

◆ Terminal prozor razvojnog okruženja:

ng g c auth/signup

ng g c auth/login

ng g c welcome

ng g c training

ng g c training/previous-training

ng g c training/current-training

ng g c training/new-training

Kreirane komponente

The image shows a screenshot of a code editor with a dark theme. On the left is the Explorer sidebar, which lists the project structure under 'ANGULAR-FITNESS-APP_V1'. The 'src' folder contains 'app', 'auth', 'login', 'signup', 'training', and 'welcome' components, each with their respective CSS, HTML, and TypeScript files. The 'welcome' component is currently selected in the Explorer. On the right is the main editor area, showing the content of 'welcome.component.html'. The code is as follows:

```
1 <p>
2   welcome works!
3 </p>
4
```

Servisi – primeri korišćenja kao simulacija pozadinske logike

- ◆ Interaktivni prototip korisničkog interfejsa bez pozadinske logike, tj. implementacije backend-a
- ◆ Pozadinska logika se simulira u pozadinskom kodu aplikacije
- ◆ Koraci:
 - ◆ Kreiranje interfejsa koji definiše strukturu objekta aplikacije
 - ◆ Definisanje servisa koji implementira funkcije čitanja, izmene i brisanja objekata
 - Testni objekti se kreiraju kao kolekcija objekata interfejsa (na primer JS niz tipa object ili TS Array)



Simulacija pozadinske logike: Slučaj rada sa nalozima korisnika

1 - Kreiranje interfejsa kao strukture objekta aplikacije sa testnim podacima

```
export interface User {  
    id: number;  
    email: string;  
    password: string;  
    firstName: string;  
    lastName: string;  
    phone: string;  
    address: string;  
    type: 'driver' | 'passenger';  
    rating: number;  
}
```

2.1 – Definisanje servisa za rad sa testnim podacima interfejsa – rad sa korisničkim nalozima

```
@Injectable()
export class UserService {
    currentUser: User = UserService.dummyUserList[0];
    static dummyUserList: Array<User> = [
        {
            id: 1,
            email: "test@test.com",
            password: "test123",
            firstName: "Test",
            lastName: "Test",
            phone: "+3816711223344",
            address: "Test Adresa 1",
            type: "driver",
            rating: 2.5,
        },
    ],
}
```

2.2 – Definisanje servisa za rad sa testnim podacima interfesa – prijava korisnika

```
import { User, UserService } from './user.service';

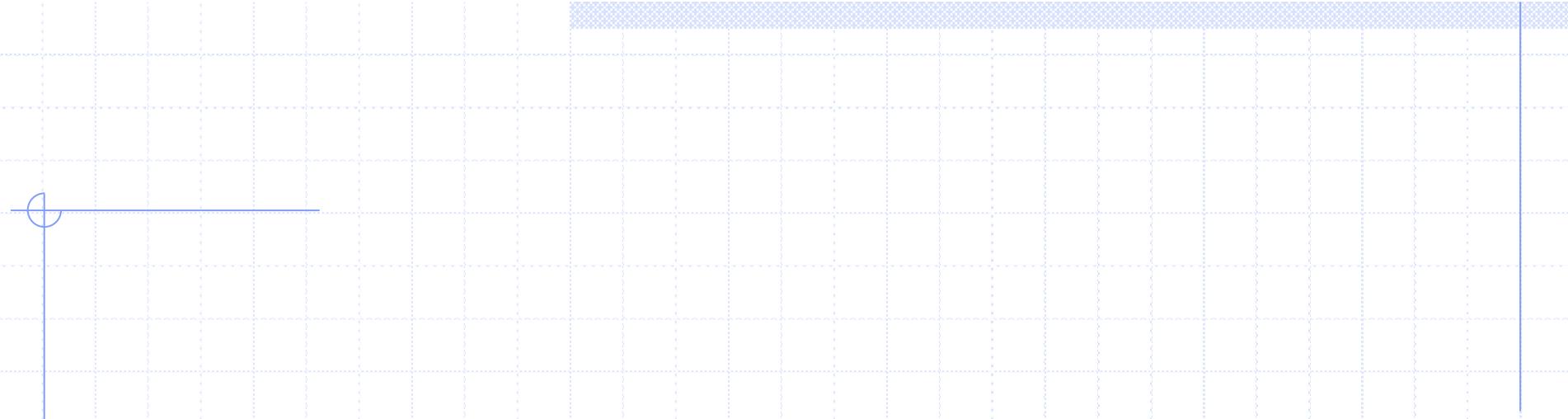
export class AuthService {
    registeredUsers: Array<User> = UserService.dummyUserList;

    getUser(userEmail: string) {
        return this.registeredUsers.find(userToFind =>
            userToFind.email == userEmail);
    }

    isPasswordCorrect(userEmail: string, password: string) {
        return this.registeredUsers.find(userToFind => (
            userToFind.email == userEmail &&
            userToFind.password == password)) != undefined;
    }
}
```

2.3 – Definisanje servisa za rad sa testnim podacima interfesa – registracija korisnika

```
registerUser(email: string, password: string, firstName: string,  
lastName: string, phone: string, address: string, type) {  
    var maxId: number = 0;  
    this.registeredUsers.forEach(user => {  
        if (maxId < user.id) {  
            maxId = user.id;  
        }  
    });  
    var id = ++maxId;  
    var user: User = {id, email, password, firstName,  
                    lastName, phone, address, type, rating: 0};  
    this.registeredUsers.push(user);  
    UserService.dummyUserList.push(user);  
  
    return user;  
}
```



Simulacija pozadinske logike: Slučaj rada sa vožnjama

1 - Kreiranje interfejsa kao strukture objekta aplikacije sa testnim podacima

```
export interface Ride {  
    id: number;  
    title: string;  
    driverId: number;  
    passengerIds: Array<number>;  
    startTime: Date;  
    startHours: number;  
    startMinutes: number;  
    startLocation: string;  
    destinationLocation: string;  
    openSlots: number;  
    status: 'open' | 'in progress' | 'finished' | 'cancelled';  
}
```

2 – Definisanje servisa za rad sa testnim podacima interfejsa – rad sa voznjama

```
@Injectable()  
export class RideService {  
  
    public dummyRideList: Array<Ride> = [  
        {  
            id: 1,  
            title: "We are going to stop at the post office on the way",  
            driverId: 1,  
            passengerIds: [],  
            startTime: new Date("2019-04-17 17:00"),  
            startHours: 17,  
            startMinutes: 0,  
            startLocation: "Danijelova 32",  
            destinationLocation: "Kumodraska 261a",  
            openSlots: 3,  
            status: 'cancelled'  
        },  
    ]};
```

2 – Definisanje servisa za rad sa testnim podacima interfejsa – čitanje vožnji

```
getAllRides() {
    return this.dummyRideList;
}

getDriverById(id: number) {
    var foundUser: User;
    UserService.dummyUserList.forEach(user => {
        if (user.id == id) {
            foundUser = user;
        }
    });
    return foundUser;
}

getRideById(id: number) {
    var foundRide: Ride;
    this.dummyRideList.forEach(ride => {
        if (ride.id == id) {
            foundRide = ride;
        }
    });
    return foundRide;
}
```

2 – Definisanje servisa za rad sa testnim podacima interfejsa – dodavanje i uklanjanje vožnji

```
createNewRide(ride: Ride) {  
    this.dummyRideList.push(ride);  
}  
  
removeRide(rideToDelete: Ride) {  
    this.dummyRideList = this.dummyRideList.filter(ride =>  
        ride.id != rideToDelete.id);  
}
```

Korišćenje servisa iz kontrolera komponente

Testni podaci kao model za tabelu

```
export class RidesComponent implements OnInit {  
    displayedColumns = ["title", "driver", "startLocation",  
    | | | | | | | | "destination", "date", "status"];  
    ridesSource = new MatTableDataSource<Ride>()  
  
    @ViewChild(MatSort) sort: MatSort;  
    @ViewChild(MatPaginator) paginator: MatPaginator;  
  
    constructor(private rideService: RideService,  
    | | | | | private userService: UserService) { }  
  
    ngOnInit() {  
        this.ridesSource.data = this.rideService.getAllRides();  
    }  
}
```

Angular Material (AM)

- ◆ Angular komponente korisničkog interfejsa koje implementariju Google Material Design specifikaciju
- ◆ **Google Material Design:** standard za kreiranje upotrebljivih Web korisničkih interfejsa
 - Komponente
 - Boje, oblici, tipografija, simboli
 - Vizuelni dizajn i raspoređivanje
- ◆ Angular Material: implementacija standarda
 - Fokus na komponente i raspoređivanje
- ◆ Kreiranje korisnih i upotrebljivih Web korisničkih interfejsa
- ◆ Olakšava i ubrzava razvoj Web korisničkih interfejsa
 - Efikasnost (brzina) – predefinisane komponente koje razvoj čine bržim
 - Efektivnost (kvalitet) – implementacija GMD kao prakse dobrog dizajna

Angular Material (AM)

◆ Sastoji se iz dva dela:

◆ Components

- Elementi korisničkog interfejsa
- Dugme
- Meni
- Tabela ...

◆ Component Dev Kit (CDK)

- Zajedničke funkcionalnosti komponenti
- Drag&Drop
- Pagination
- Layout
- Scroll ...

AM Components

<https://material.angular.io/components/categories>

(izborom kategorije dobijaju se konkretni elementi)

Form Controls

Controls that collect and validate user input.

Navigation

Menus, sidenavs and toolbars that organise your content.

Layout

Essential building blocks for presenting your content.

Buttons & Indicators

Buttons, toggles, status and progress indicators.

Popups & Modals

Floating components that can be dynamically shown or hidden.

Data table

Tools for displaying and interacting with tabular data.

Komponente formi (Form Controls)

(izborom komponente dobija se opis izabrane komponente)

Autocomplete



Suggests relevant options as the user types.

Checkbox



Captures boolean input with an optional indeterminate mode.

Datepicker



Captures dates, agnostic about their internal representation.

Form field

Wraps input fields so they are displayed consistently.

Input



Enables native inputs to be used within a Form field.

Radio button



Allows the user to select one option from a group.

Select



Allows the user to select one or more options using a dropdown.

Slider



Allows the user to input a value by dragging along a slider.

Slide toggle



Captures boolean values as a clickable and draggable switch.

Komponente navigacije (Navigation)

Menu



A floating panel of nestable options.

Sidenav



A container for content that is fixed to one side of the screen.

Toolbar



A container for top-level titles and controls.

Komponente rasporedivanja (Layout)

Card



A styled container for pieces of itemized content.

Divider

A vertical or horizontal visual divider.

Expansion Panel

A container which can be expanded to reveal more content.

Grid list



A flexible structure for presenting content items in a grid.

List



Presents conventional lists of items.

Stepper

Presents content as steps through which to progress.

Tabs



Only presents one view at a time from a provided set of views.

Tree

Presents hierarchical content as an expandable tree.

Komponente dugmadi i indikatora (Buttons & Indicators)

Button



An interactive button with a range of presentation options.

Chips



Presents a list of items as a set of small, tactile entities.

Progress bar



A linear progress indicator.

Button toggle



A groupable on/off toggle for enabling and disabling options.

Icon



Renders a specified icon.

Ripples

Directive for adding Material Design ripple effects

Badge

A small value indicator that can be overlaid on another object.

Progress spinner



A circular progress indicator.

Komponente dijaloga (Popups & Modals)

1

Bottom Sheet

A large interactive panel primarily for mobile devices.

Dialog



Snackbar



Tooltip



Displays floating content when an object is hovered.

Komponente table (Popups & Modals)

Paginator

Controls for displaying paged data.

Sort header

Allows the user to configure how tabular data is sorted.

Table

A configurable component for displaying tabular data.

Dokumentacija AM komponente

◆ Opis svake komponente je dat u tri sekcije:

◆ **Pregled (overview)**

- Opis komponente sa primerima korišćenja

◆ **API**

- Definicija komponente
- Opisi atributa (properties)

◆ **Primeri (examples)**

- Primeri korišćenja komponente
- HTML, TS i CSS
- Mogućnost editovanja u StackBlitz online razvojnom okruženju

Primer dokumentacije AM komponente

OVERVIEW API EXAMPLES

<mat-radio-button> provides the same functionality as a native <input type="radio"> enhanced with Material Design styling and animations.

Basic radios

Option 1 Option 2

All radio-buttons with the same name comprise a set from which only one may be selected at a time.

Radio-button label

Contents

- Radio-button label
- Radio groups
- Use with @angular/forms
- Accessibility

AM - instalacija

- ◆ Podrška za AM se instalira na nivou konkretnog projekta

```
npm install --save @angular/material @angular/cdk  
@angular/animations
```

--save ažurira sadržaj package.json sa zavisnostima (verzije instaliranih biblioteka)

@angular/material – biblioteka AM komponenti

@angular/cdk – biblioteka zajedničkih funkcija komponenti (component dev kit)

@angular/animations – predefinisane animacije komponenti (na primer, efekat talasa na klik dugmeta)

Instalirane biblioteke – package.json

```
"dependencies": {  
    "@angular/animations": "^7.2.12",  
    "@angular/cdk": "^7.3.7",  
    "@angular/common": "~7.2.0",  
    "@angular/compiler": "~7.2.0",  
    "@angular/core": "~7.2.0",  
    "@angular/flex-layout": "^7.0.0-beta.24",  
    "@angular/forms": "~7.2.0",  
    "@angular/material": "^7.3.7",  
    "@angular/platform-browser": "~7.2.0",  
    "@angular/platform-browser-dynamic": "~7.2.0",  
    "@angular/router": "~7.2.0",  
    "core-js": "^2.5.4",  
    "hammerjs": "^2.0.8",  
    "rxjs": "~6.3.3",  
    "tslib": "^1.9.0",  
    "zone.js": "~0.8.26"  
},
```

Korišćenje AM komponenti

- ◆ Svaka AM komponenta ima svoju implementaciju
 - MatButtonModule, MatIconModule, MatTableModule
 - Nalaze su u @angular/material biblioteci
- ◆ Da bi se komponenta koristila mora se eksplisitno importovati u aplikaciju
- ◆ Da bi AM komponenta bila vidljiva u čitavoj aplikaciji **preporuka** je sledeća:
 1. Kreira se modul za registraciju AM komponenti u src/app direktotrijumu projekta (material.module.ts)
 2. Modul se zatim registruje u korenom modulu aplikacije (app.module.ts)
 3. Prilikom prvog korišćenja AM komponente, potrebno je registrovati komponentu u material.module.ts
- ◆ Na ovaj način AM komponenta je vidljiva u okviru čitave aplikacije, tj. ne mora se importovati u svakoj komponenti aplikacije koja koristi dati element interfejsa !

material.module.ts – modul za registraciju AM komponenti

Kreira se programski u src/app folderu
AM komponente se registruju kao ispod:

- ◆ import naredba
- ◆ imports i exports atributi @NgModule dekoratora

```
import { NgModule } from '....../node_modules/@angular/core';
import { MatButtonModule } from '@angular/material';

@NgModule ({
  imports: [MatButtonModule],
  exports: [MatButtonModule]
})

export class MaterialModule {}
```

app.module.ts (korenji modul)



Nako toga se kreirani modul registruje u korenom modulu aplikacije:

- ◆ **import** naredba u zaglavlju
- ◆ **imports** atribut **@NgModule** dekoratora

```
@NgModule({  
  declarations: [  
    AppComponent,  
    SignupComponent,  
    LoginComponent,  
    WelcomeComponent,  
    TrainingComponent,  
    PreviousTrainingComponent,  
    CurrentTrainingComponent,  
    NewTrainingComponent  
  ],  
  imports: [  
    BrowserModule,  
    BrowserAnimationsModule,  
    MaterialModule,  

```

Teme

- ◆ AM tema je skup boja koje će biti primenjene na komponente korisničkog interfejsa Angular aplikacije
- ◆ Generišu se statički, za vreme prevodenja aplikacije
- ◆ Tema se sastoji od pet paleta boja za vizuelno označavanje različitih stanja u kojima može biti komponenta korisničkog interfejsa:
 - ◆ **primarna** paleta (*primary*) - najčešće korišćene boje
 - ◆ **naglašena** (*accent*) paleta - boje za interaktivne elemente,
 - ◆ paleta **upozorenja** (*warn*) - boje za prikazivanje grešaka,
 - ◆ **frontalna** paleta (*foreground*) - boje za tekst i ikone (elementi u prvom planu),
 - ◆ **pozadinska** paleta (*background*) - boje za pozadine elementa

Tema – šematski prikaz sastavnih paleta

Primary Palette

Most widely used color across all screens

Accent Palette

Used on interactive elements

Warn Palette

Convey warnings

Foreground Palette

For text and icons

Background Palette

For backgrounds

Predefinisane teme

- ◆ Četiri predefinisane teme (naziv odgovara primarnoj boji ili paleti):
 - ◆ `deeppurple-amber.css`,
 - ◆ `indigo-pink.css`,
 - ◆ `pink-bluegrey.css`,
 - ◆ `purple-green.css`
- ◆ Nalaze se u `@angular/material/prebuilt-themes` biblioteci
- ◆ Da bi se koristile, tj, bile primenjene na elemente korisničkog interfejsa moraju se navesti eksplicitno u `styles.css`:
`@import '~@angular/material/prebuilt-themes/indigo-pink.css';`

Nazivi predefinisanih tema
odgovaraju nazivima
paleta boja
Google Material Design

deeppurple-amber.css

Deep purple 50	#EDE7F6	Amber 50	#FFF8E1
100	#D1C4E9	100	#FFECB3
200	#B39DDB	200	#FFE082
300	#9575CD	300	#FFD54F
400	#7E57C2	400	#FFCA28
500	#673AB7	500	#FFC107
600	#5E35B1	600	#FFB300
700	#512DA8	700	#FFA000
800	#4527A0	800	#FF8F00
900	#311B92	900	#FF6F00
A100	#B388FF	A100	#FFE57F
A200	#7C4dff	A200	#ffd740
A400	#651FFF	A400	#ffc400
A700	#6200EA	A700	#ffab00

Nazivi predefinisanih tema
odgovaraju nazivima
paleta boja
Google Material Design

indigo-pink.css

Indigo 50	#E8EAF6	Pink 50	#FCE4EC
100	#C5CAE9	100	#F8BBDO
200	#9FA8DA	200	#F48FB1
300	#7986CB	300	#F06292
400	#5C6BC0	400	#EC407A
500	#3F51B5	500	#E91E63
600	#3949AB	600	#D81B60
700	#303F9F	700	#C2185B
800	#283593	800	#AD1457
900	#1A237E	900	#880E4F
A100	#8C9EFF	A100	#FF80AB
A200	#536DFE	A200	#FF4081
A400	#3D5AFE	A400	#F50057
A700	#304FFE	A700	#C51162

Nazivi predefinisanih tema
odgovaraju nazivima
paleta boja
Google Material Design

pink-bluegrey.css

Pink 50	#FCE4EC	Blue Gray 50	#ECEFF1
100	#F8BBD0	100	#CFD8DC
200	#F48FB1	200	#B0BEC5
300	#F06292	300	#90A4AE
400	#EC407A	400	#78909C
500	#E91E63	500	#607D8B
600	#D81B60	600	#546E7A
700	#C2185B	700	#455A64
800	#AD1457	800	#37474F
900	#880E4F	900	#263238
A100	#FF80AB		
A200	#FF4081		
A400	#F50057		
A700	#C51162		

Nazivi predefinisanih tema
odgovaraju nazivima
paleta boja
Google Material Design

purple-green.css

Purple 50	#F3E5F5	Green 50	#E8F5E9
100	#E1BEE7	100	#C8E6C9
200	#CE93D8	200	#A5D6A7
300	#BA68C8	300	#81C784
400	#AB47BC	400	#66BB6A
500	#9C27B0	500	#4CAF50
600	#8E24AA	600	#43A047
700	#7B1FA2	700	#388E3C
800	#6A1B9A	800	#2E7D32
900	#4A148C	900	#1B5E20
A100	#EA80FC	A100	#B9F6CA
A200	#E040FB	A200	#69F0AE
A400	#D500F9	A400	#00E676
A700	#AA00FF	A700	#00C853

Predefinisane teme

- ◆ Implementacije predefinisanih tema se mogu videti u projektu
- ◆ Generišu se sa AM instalacijom

node_modules/@angular/material/prebuilt-themes

```
◀ prebuilt-themes
# deeppurple-amber.css
# indigo-pink.css
# pink-bluegrey.css
# purple-green.css
```

Teme – primer korišćenja

- ◆ Jednom kada izaberemo temu, ona će se automatski primenjivati na komponente korisničkog interfejsa
- ◆ AM omogućava bojenje komponente tako što se *color* atributu komponente dodeli jedna od tri vrednosti - *primary*, *accent* ili *warn*
- ◆ Primer teme deeppurple-amber.css

```
<button mat-raised-button color="primary">Primary</button>
<button mat-raised-button color="accent">Accent</button>
<button mat-raised-button color="warn">Warn</button>
```

Primary

Accent

Warn

Animacije AM komponenti

- ◆ Angular omogućava korišćenje predefisanih animacija kao različitih oblika tranzicija (pomeraji i promene boja, oblika i pozicija)
- ◆ One su definisane kao modul (BrowserAnimationsModule) koji je potrebno registrovati u koreni modul (app.module.ts):

```
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
```

```
imports: [
  BrowserModule,
  MaterialModule,
  RoutingModule,
  BrowserAnimationsModule],
```

Dugme (button) 1/4

Korišćenje:

```
import {MatButtonModule} from '@angular/material';
imports i exports atributi @NgModul dekoratora
(material.module.ts)
```

Varijante:

Istaknuto dugme (*raised button*): <mat-raised-button>

```
<button mat-raised-button>Basic</button>
<button mat-raised-button color="primary">Primary</button>
<button mat-raised-button color="accent">Accent</button>
<button mat-raised-button color="warn">Warn</button>
<button mat-raised-button disabled>Disabled</button>
```

Basic

Primary

Accent

Warn

Disabled

Dugme (button) 2/4

Ovičeno dugme (*stroked button*): <mat-stroked-button>

```
<button mat-stroked-button>Basic</button>
<button mat-stroked-button color="primary">Primary</button>
<button mat-stroked-button color="accent">Accent</button>
<button mat-stroked-button color="warn">Warn</button>
<button mat-stroked-button disabled>Disabled</button>
```

Basic

Primary

Accent

Warn

Disabled

Ravno dugme (*flat button*): <mat-flat-button>

```
<button mat-flat-button>Basic</button>
<button mat-flat-button color="primary">Primary</button>
<button mat-flat-button color="accent">Accent</button>
<button mat-flat-button color="warn">Warn</button>
<button mat-flat-button disabled>Disabled</button>
```

Basic

Primary

Accent

Warn

Disabled

Dugme (button) 3/4

Dugme ikona (*icon button*): <mat-icon-button> i <mat-icon>

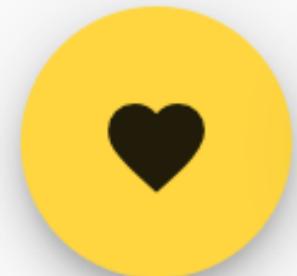
```
<button mat-icon-button>
  <mat-icon>favorite</mat-icon>
</button>
<button mat-icon-button color="primary">
  <mat-icon>favorite</mat-icon>
</button>
<button mat-icon-button color="accent">
  <mat-icon>favorite</mat-icon>
</button>
<button mat-icon-button color="warn">
  <mat-icon>favorite</mat-icon>
</button>
<button mat-icon-button disabled>
  <mat-icon>favorite</mat-icon>
</button>
```



Dugme (button) 4/4

Kružno dugme (*fab button*): <mat-fab> <mat-mini-fab>

```
<button mat-fab>Basic</button>  
<button mat-fab color="primary">Primary</button>  
<button mat-fab color="accent">Accent</button>  
<button mat-fab color="warn">Warn</button>  
<button mat-fab disabled>Disabled</button>  
  
<button mat-fab>  
  <mat-icon>favorite</mat-icon>  
</button>
```



Dugme alternativa

- ◆ Za svaki od podtipova umesto <button> se može koristiti <a>
- ◆ Na primer, za linkove ruta

```
<a mat-raised-button color="primary" routerLink=".">Link</a>
```



```
<a mat-fab routerLink=".">Link</a>
```



Ikone (icons)

- ◆ Predstavljaju grafičke simbole kao vektorske objekte (svg datoteke)
`<mat-icon>ime_simbola</mat icon>`
- ◆ Simboli se mogu instalirati lokalno, ili se može direktno referencirati Google Material Design kolekcija ikona
- ◆ Mogu se bojiti kao i dugmad (color = {primary | warn | accent})
- ◆ Upotrebe:
 - ◆ Dekorativna: informacija bez interaktivnosti (legenda)
 - ◆ Interaktivna: dugme ikona koja pokreće akciju
 - ◆ Indikator: poruka ili status izvršenja akcije (obično uz tekst)

Ikone (icons)

Korišćenje:

```
import {MatIconModule} from '@angular/material';
imports i exports atributi @NgModul dekoratora
(material.module.ts)
```

Interaktivna upotreba: <mat-icon> kao dete <button> ili
<a> elementa

Indikator: kombinacija sa inline kontejnerom za
ispisivanje poruke (kao deo bloka teksta)

Korišćenje Google Material kolekcije ikona

U index.html stranicu dodati referencu na kolekciju:

```
<link href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">
```

Google Material Design kolekcija ikona

<https://material.io/tools/icons/>

Simboli se referenciraju po nazivu

Mogu se koristiti i lokalno (download)

The screenshot shows a grid of Google Material Design icons. On the left, there is a sidebar with filtering options:

- Filter by name:** A search bar.
- THEMES:**
 - Filled
 - Outlined
 - Rounded
 - Two-Tone
 - Sharp
- CATEGORIES:** A dropdown menu set to "All".

The main area displays a 5x10 grid of icons, each with its name below it. The icons are categorized by theme:

- Filled:** 3d_rotation, accessibility, accessibility..., accessible, accessible_for..., account_balance, account_box, account_circle, add_shopping...
- Outlined:** alarm, alarm_add, alarm_off, alarm_on, all_inbox, all_out, android, announcement, arrow_right, aspect_ratio
- Rounded:** assessment, assignment, assignment_in..., assignment_la..., assignment_re..., assignment_re..., assignment_tu..., autorenew, backup, book
- Two-Tone:** bookmark, bookmark_bord..., bookmarks, bug_report, build, cached, calendar_today, calendar_view..., camera_enhanc..., card_gif...
- Sharp:** card_membership, card_travel, change_history, check_circle, check_circle..., chrome_reader..., class, code, commute, compare_arrow...

Ikone (icons)

```
<button mat-icon-button color="accent">  
    <mat-icon>home</mat-icon>  
</button>
```



Ako se online kolekcija ikona pretraži po ključnoj reči 'home' dobiće se simbol iznad



home

AM i teme podsetnik za rad

◆ AM komponente:

- ◆ Kreirati modul za registraciju AM komponenti (material.module.ts)
- ◆ Registrovati modul u korenom modulu aplikacije (app.module.ts)
- ◆ Svaku AM komponentu koja se želi koristiti je potrebno registrovati u material.module.ts

◆ Teme:

- ◆ Definisati globalnu temu na nivou aplikacije u styles.css
 - Izabrati postojeću ili kreirati svoju

Sažetak

- ◆ Rutiranje
- ◆ Kreiranje i konfiguracija projekta
- ◆ Servisi
 - Slučajevi korišćenja kao simulacija pozadinske logike
- ◆ Angular Material komponente korisničkog interfejsa:
- ◆ Instalacija
- ◆ Kolekcija komponenti
- ◆ Teme
- ◆ Korišćenje (dugme, simboli)

Materijali

- ◆ **Upustvo za kreiranje projekta:**

Angular – razvojno okruženje i kreiranje projekta.pdf

- ◆ **Instalacije:**

Instalacije.pdf

- ◆ **Servisi:**

auth.service.ts

ride.service.ts

user.service.ts

- ◆ **Projekat:**

projekat_primer.zip (kreiran po uputsvu)

Linkovi ka dokumentaciji i primerima

- ◆ Google Material Design: <https://material.io/design/>
- ◆ Angular Material: <https://material.angular.io/>
- ◆ Angular Material komponente:
<https://material.angular.io/components/categories>
- ◆ Angular Material teme:
<https://material.angular.io/guide/theming>
- ◆ Google Material Design paleta boja:
<https://material.io/design/color/the-color-system.html#tools-for-picking-colors>
- ◆ Primeri dugmeta za promenu tema i boja:
<https://stackblitz.com/angular/dnbnrjyaxaeqy?file=app%2Fbutton-types-example.html>
- ◆ Primer za simbole ikona:
<https://stackblitz.com/angular/yonlxjxpgjd?file=index.html>
- ◆ Google Material Design kolekcija ikona:
<https://material.io/tools/icons/>

Rutiranje

Konfiguracija projekta

Servisi: simulacija pozadinske

logike

Angular Material uvod

Dr. Mlađan Jovanović
mjovanovic@singidunum.ac.rs