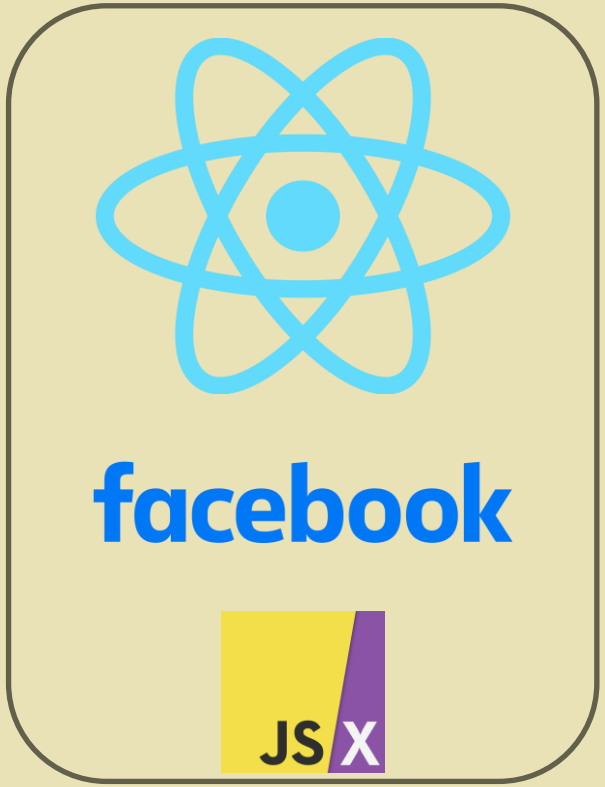
Programiranje internet aplikacija

Elektrotehnički fakultet, Univerzitet u Beogradu

2024/2025

Uvod

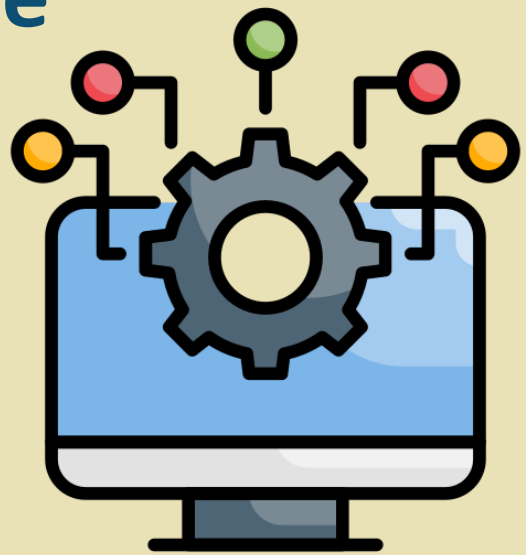
- Front-end JavaScript framework
- Najpoznatiji:



Angular

- TypeScript – proširuje JavaScript tipovima
- Glavni koncepti
  - NgModules
  - Komponente
  - Two-way binding
  - Servisi
  - Dependency injection
  - Rutiranje
- Single page aplikacija

Instalacija, kreiranje i pokretanje aplikacije



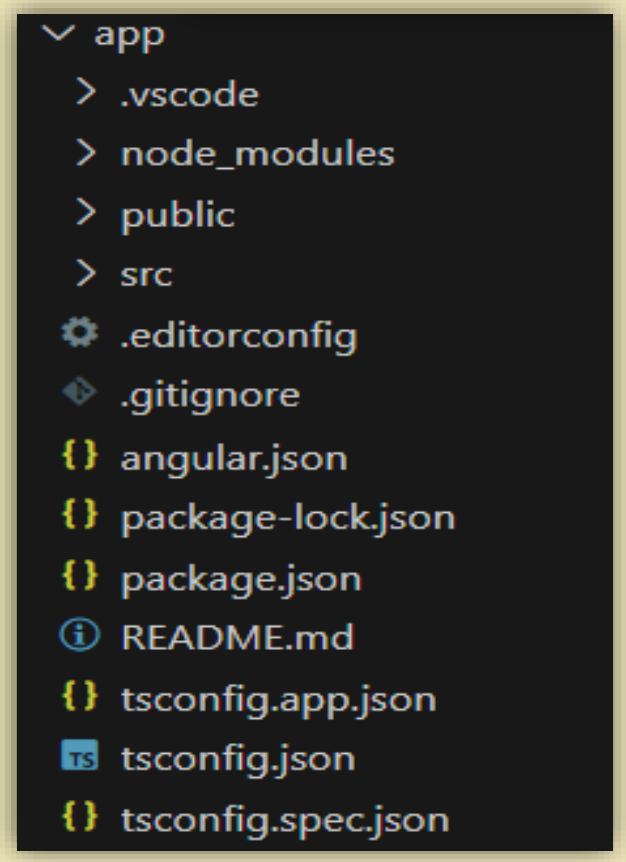
- Node.js (20.17.0)
  - https://nodejs.org/dist/v20.17.0/
  - Node package manager

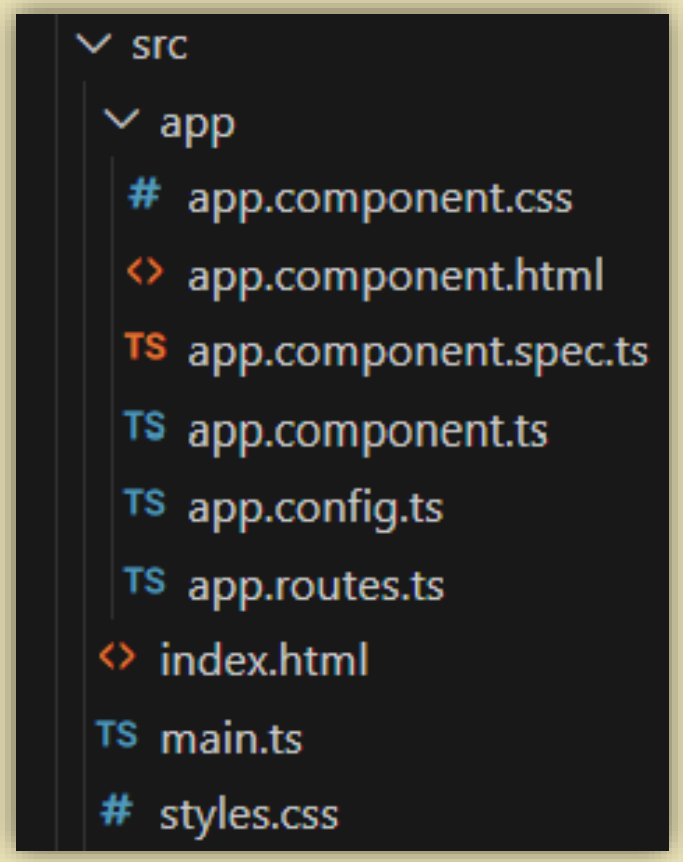
```
npm install -g @angular/cli@18.2.5
```

```
ng new naziv_aplikacije
```
- style – CSS
- Do you want to enable Server-Side Rendering (SSR) and Static Site Generation (SSG/Prerendering)? No

```
npm start
```

Struktura aplikacije





Interpolacija

```ts
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  standalone: true,
  imports: [],
  templateUrl: './app.component.html',
  styleUrl: './app.component.css'
})
export class AppComponent {
  appTitle = 'app';
}
```

```html
<h3>Welcome {{appTitle}}</h3>
```

Kreiranje komponenti, servisa i rutiranje

- Kreiranje komponenti

ng generate component component_name
(ng g c component_name)

- Kreiranje servisa

ng generate service service_name
(ng g s service_name)

- Rutiranje

```
import { Routes } from '@angular/router';
import { AboutComponent } from './about/about.component';
import { BooksComponent } from './books/books.component';
import { WritersComponent } from './writers/writers.component';

export const routes: Routes = [
    {path: "about", component: AboutComponent},
    {path: "books", component: BooksComponent},
    {path: "writers", component: WritersComponent}
];
```

```
import { Component } from '@angular/core';
import { RouterLink, RouterOutlet } from '@angular/router';

@Component({
  selector: 'app-root',
  standalone: true,
  imports: [RouterLink, RouterOutlet],
  templateUrl: './app.component.html',
  styleUrl: './app.component.css'
})
export class AppComponent {
  title = 'app';
}
```

```
<h3>Welcome {{title}}</h3>
<a routerLink="/about">About</a> |
<a routerLink="/books">Books</a> |
<a routerLink="/writers">Writers</a>
<router-outlet></router-outlet>
```

Direktive

Klase koje definišu strukturu i ponašanje

- **Direktiva komponenta** – direktive sa šablonom
- **Strukturalne direktive** – direktive koje manipulišu elementima u DOM-u
  - *ngFor, *ngIf, *ngSwitch
  - @for, @if, @else, @switch
  - Jedna strukturalna direktiva po elementu
- **Atributske direktive** – direktive koje manipulišu izgledom elemenata u DOM-u

Filteri (*pipes*)

Specijalni operatori koji omogućavaju transformaciju podataka

Ugrađeni:

- CurrencyPipe
- DatePipe
- LowerCasePipe
- ...

```
<p>
        The event will occur on
        {{ scheduledOn | date | uppercase }}.
</p>
```

```
<p>
        The event will occur at
        {{ scheduledOn | date:'shortDate' }}.
</p>
```

Mogu se kreirati i filteri za specifične potrebe

One-way binding

- **Interpolacija**
- **Property binding**

  <img alt="item" [src]="itemImageUrl"> HTML

  itemImageUrl = "img/newImage.jpg"; TS
- **Attribute binding**

  <button type="button" [attr.aria-label]="actionName">

  {{actionName}} with Aria

  </button>
- **Class and style binding**

  [class.sale]="onSale" true, false
- **Event binding**

  <button (click)="onSave()">Save</button>

from data source to view target

from view target to data source

Prosleđivanje podataka u komponentu

```ts
import { Component, Input } from '@angular/core';
import { Book } from '../models/book';

@Component({
  selector: 'app-bookdetails',
  standalone: true,
  imports: [],
  templateUrl: './bookdetails.component.html',
  styleUrl: './bookdetails.component.css'
})
export class BookdetailsComponent {
  @Input() myBook: Book = new Book()
}
```

```
@for(book of allBooks; track book.name){
    <app-bookdetails [myBook]="book"></app-bookdetails>
    <hr>
}
```

Forme

- reactive vs **template-driven**

```
import { FormsModule }
from '@angular/forms'
```

```
<form>
    <input type="text" name="param" [(ngModel)]="searchParam">
    <button (click)="search()">Search</button>
</form>
```

Two-way binding

Povezivanje sa backend-om

- **ProvideHttpClient**

```
import { provideHttpClient } from
'@angular/common/http';

export const appConfig: ApplicationConfig = {
providers: [provideZoneChangeDetection({
eventCoalescing: true }), provideRouter(routes),
provideHttpClient()]};
```

- **HttpClient**

```
import { HttpClient }
from '@angular/common/http’


private http = inject(HttpClient)
```

- Uz pomoć HttpClient-a šaljemo HTTP zahteve ka backend-u, a kao odgovor dobijamo Observable
- Observable se izvršava tek kada se na njega subscribe-ujemo

Guards

```
export const authenticationGuard : CanActivateFn = (route, state) => {
        const oauthService: AuthService = inject(AuthService);

        if (oauthService.hasAccess() ) {
                return true;
        }

        return false;
        };
}
```

ng generate guard guard_name
(ng g g guard_name)

```
const routes: Route[] = [
  {
        path: 'home',
        component: HomeComponent,
        canActivate: [authenticationGuard]
  }
]
```

HVALA NA PAŽNJI!