# Struktura angular-a

## Single page app vs multiple-page application

SPA is fast, as most resources (HTML+CSS+Scripts) are only loaded once throughout the lifespan of application and rendered trough javascript client(client side rendering). Only data is transmitted back and forth. Lak razvoj I debug aplikacije. Mana je teska SEO optimizacija zato sto je strana koju ucitavamo na backendu prazna , google na primer ne moze da pretrazi sadrzaj nase strane jer se sadrzaj dodaje naknadno pomocu JS-a.

Kod MPA(Server side rendering) na backendu postoji ruta za svaku stranicu(staticki html sa javascript kodom)

## pros and cons of Angular compared to React?

React je biblioteka a angular razvojno okruzenje.

Angular dolazi sa preinstaliranim feature-ima kao sto su RXjS, rutiranje, animacije, dependency injection...

Angular se zasnima na MVC(model,view,controler) arhitekturi, a React na Komponentama za koje koristi JSX(JavaScript XML, allows us to write HTML in React).

React ima bolje SEO u odnosu na angular zbog SSR(Server side rendering)-a.

Angular koristu two-way data binding can affect performance, as it constantly checks for changes in both the model and the view.

React’s one-way data binding and the use of a virtual DOM contribute to better performance, as it only updates the parts of the DOM that have changed.

A comparison of a model binding

Description automatically generated

## Struktura Angular Projekta

U root folderu se nalaze svi konfiguracioni fajlovi za typescript, biblioteke i angular.

Unutar src foldera se nalaze globalni css fajl, glavni html fajl , main.ts fajl i app folder.

Main.ts je prvi fajl koji se izvrsava.

U app folderu se nalazi root componenta

## Module

Modul predstavlja kolekciju komponenti,servisa,pipe-ova...

AppModule je glavni root module.

Od angulara 17 se ne kreira automatski vec je postao opciona stavka. Ovo je postignuto pomocu standallone komponenti

A screen shot of a computer program

Description automatically generated

**Declarations:** Lista deklarisanih komponenata unutar modula. U Koliko pokusamo da koristimo komponentu bez da je prethodno deklarisemo docice od greske.

**Imports:** sluzi da importujemo druge module koji su nam potrebni za rad aplikacije

**Providers:** koristi se za dependecy injection.

**Bootstrap:** the root component that Angular creates and inserts into the index.html host web page.

### What is ForRoot & ForChild in Angular?

ForRoot je staticna funkcija unutar modula koja se koristi za dodavanje konfiguracije i servisa unutar modula. Dobar primer je dodavanje ruta u RoutingModulu. Ova funkcija je depricated i sada se koristi @Injectable({providedIn: 'root'}) dekorator.

A computer code with pink letters

Description automatically generated

Primer custom forRoot-a u custom modulu

A screen shot of a computer

Description automatically generated

A computer code with colorful text

Description automatically generated

A black background with red and yellow text

Description automatically generated

A screen shot of a computer program

Description automatically generated

In Angular, the forChild method is used to define child routes within a module.

## Komponente

Osnovni blok gradnje u angular-u.

Kreira se pomocu komande ng new c NazivKomponente.

Komponenta se sastoji od Dekoratora i ts klase. Dekorator u sebi sadrzi metapodatke kao sto su:

-selector : naziv komponente u html-u

-template: html komponente

-styleUrl: css komponente

-providers: lista provider-a za ovu komponentu i njenu decu

A screen shot of a computer code

Description automatically generated

### Component vs Directive - what is the difference?

A red rectangular sign with black text

Description automatically generatedA yellow rectangular sign with black text

Description automatically generated

Component directive su komponente koje u sebi sadrze template odnosno izgled elementa.

Atribute directive se koristi za menjanje izgleda ili ponasanja DOM elemenata. ngStyle ngClass

Strukturne direktive se koriste za prikazivanje/skrivanje dom ele. ngIf ngFor

### Standalone komponente

Omogucavaju nam da definisemo komponente bez upotrebe modula. Ako zelimo da standalone komponentu koristimo u drugoj standalone komponenti potrebno je da je importujemo kod dekoratora komponente.

### Lifecycle

Kada angular aplikacija zapocne sa radom ona prvo renderuje root komponent-u. Zatim renderuje njegovu decu pa decu od dece.

Zivotni ciklus komponentes se moze pratiti uz pomoc sledecih metoda:

* ngOnInit: kada se prvi put inicijalizuje
* ngOnChanges: kada se izvrsi event koji rezultira promenom DOM-a(input,httpreq)
* ngOnDestroy: kada se komponenta brise
* ngAfterOnInit: kada komponenta zavrsi sa renderovanjem

### Sharing data between components in Angular

A group of black text

Description automatically generated

### Constructor vs NgOnInit in Angular

Konstruktor se prvi poziva i ngOnInit se poziva kada se konstruktor zavrsi.

## Hierarchical Injection

A diagram of components

Description automatically generated

Ako komponenti C dodelimo(providers[]) neki element ista instanca tog elementa ce biti dostupna i u komponentama c1 i c2.

Ako koristimo provider u child komponentama onda ce one sve tri komponente imati razlicite instance.

A screenshot of a computer

Description automatically generated

Ako se instanca dependency-a ne nadje kad se stigne do root componente onda se prelazi na pretrazivanje unutar module injector-a. U koliko se ni tu ne pronadje odgovarajuca instanca onda se baca greska.

A diagram of a diagram

Description automatically generated with medium confidenceA white background with black text

Description automatically generated

## ViewChild and ViewChildren in Angular?

Dekoratori koji se koriste za pristupanje DOM elementima kao i metodama i propetiima child komponenata.

A computer code on a white background

Description automatically generated

A computer screen shot of a computer code

Description automatically generated

## Difference between ng-template, ng-container, and ng-content

<**ng**-**template**> is a **template** element that **Angular** uses with structural directives ( \*ngIf, \*ngFor, [ngSwitch], and custom directives). Ng-templeate se ne renderuje u koliko se ne pozove pomocu direktiva.

A screen shot of a computer

Description automatically generated

**ng-container** is an extremely simple directive that allows you to group elements in a template that doesn’t interfere with styles or layout because Angular doesn’t put it in the DOM

A screenshot of a computer code

Description automatically generated

Ng-content se koristi za injectovanje html-a iz parenta u child komponentu.

A screen shot of a computer code

Description automatically generated

A screenshot of a computer program

Description automatically generated

ngTemplateOutlet is an Angular directive that allows you to dynamically render a template within a component's content. It allows you to pass in a template as input to a component, which can be used to render the content of the component in a customized way.

A screenshot of a computer code

Description automatically generated

## JIT and AOT

Angular provides two compilation techniques, AOT(Ahead of Time) and JIT(Just-in-Time)

AOT je server side rendering, za produkciju

JIT client side rendering, best when your application is in local development

## Data Binding in Angular

String binding {{}}

Property [src]=““

Event (click)=“onClick()“

TwoWay binding [(ngModel)]=““

## Pipe

Koriste se za transformaciju podataka

{{ promenljiva | async}} async pipe primer, prednost je sto se automatsku unsubscibe-uje kad se komponenta izbrise.

## Routing

Izvrsava se u routing modulu. Svaka ruta ima svoju putanju,komponentu,decuRute,guardOvi,Provideri

## Difference between ngif and hidden

ngIf ne redneruje komponentu a hidden renderuje

## DOM

Document Object Model (DOM) is the object-oriented representation of an HTML or XML document.

Omogucava js da pristupi nekom elementu.

## Bubling

Event Bubbling is a concept in the DOM (Document Object Model). It happens when an element receives an event, and that event bubbles up (or you can say is transmitted or propagated) to its parent and ancestor elements in the DOM tree until it gets to the root element.A screenshot of a computer

Description automatically generated

# Zone.js

Zone.js je biblioteka koju angular koristi za change detection mehanizam. Zone.js handluje asinhrone dogadjaje kao sto su:

* DOM events (click, hover over, etc.)
* AJAX requests
* Timers (setTimer(), setInterval())

If any of these events occur in your Angular app, Zone.js will cause change detection to run.

## ngZone in Angular?

Omogucava nam da pokrenemo kod izvan angular change detection-a. Ovime se problem sa optimizacijom. Primer dole.

A black screen with white text

Description automatically generated

A screen shot of a computer code

Description automatically generated

## What are change detection and onPush in Angular?

change detection is **a built-in framework feature that ensures the automatic synchronization between the data of a component and its HTML template view. Za ovo se koristi Zone.js handluje**

* DOM events (click, hover over, etc.)
* AJAX requests
* Timers (setTimer(), setInterval())

If any of these events occur in your Angular app, Zone.js will cause change detection to run.

Postje 2 strategije: default I onPush

Default strategija poredi vrednosti pre I posle event-a I onda prolazi kroz sve komponente od root-a pa na dole I utvrdjuje koja komponenta treba da se rerenderuje.

OnPush treba da obezbedi bolje performanse tako sto komponente oznacava kao dirty(doslo je do promena) samo u koliko je doslo po promene reference u @input propertijima date komponente.

In the **Default strategy**, whenever any data to @Input() decorated properties are changed, Angular runs the change detector to update the view. In the **onPush** strategy, Angular runs change detector only when a **new reference** is passed to the @Input() decorated properties.

Bez obzira na strategiju event unutar komponente ce dovesti do rerenderovanja komponente.

OnPush ne unice na observable-e

A diagram of a company's component

Description automatically generated

A diagram of components and components

Description automatically generated

A screen shot of a computer

Description automatically generated

## Signal

A **signal** is a wrapper around a value that notifies interested consumers when that value changes. Signals can contain any value, from primitives to complex data structures.

zone.js triggers the Change Detection whenever a DOM Event happens or an asynchronous task ends.

The Change Detection must go through the complete component tree and search for changes. If it detects one, it updates the affected DOM node.

This is not very performant because the Change Detection even runs when there is no change at all.

Pomocu onpush strategije mozemo da optimizujemo program tako sto ce sada change detection pregledati samo komponente nad kojima su se desile izmene i njihove roditelje.

A diagram of a tree

Description automatically generatedU ovom primeru kada je default strategija dolazi do provere i svih komponenata uprkos tome sto je doslo do promene samo na komponenti 13. Kada koristimo onPush strategiju provera ce se izvrsiti samo nad komponentama 1,3,7 i 13.

Ovo mozemo dodatno poboljsati pomocu signala(local change detection). Ako kod komponente 13 koristimo signal umesto observable-a onda ce se change detection izvrsiti samo nad komponentom 13.

# RxJS?

Biblioteka za kreiranje asinhronih i event-base programa pomocu data stream-ova(observable).

Angular se zasniva na RxJS-u

## Promise vs observable

Promis handluje jedan event kada asinhrona operacija uspe ili ne.

Observable se ponasa kao stream tako omogucava da handluje 0 ili vise enent-ova. Observable poseduje operatore kao map, reduce za modifikovanje podataka iz stream-a. lazy’ – this means that it won’t be executed at the moment of defining the stream, but when the subscription is created

Primer koriscenja promise za HTTP zahteve a observable za onclick event

## How to transform data in RxJS?

Podaci iz observable-a se transformisu pomocu pipe funkcije

A screenshot of a computer code

Description automatically generated

## How filter works in RxJS?

A computer screen shot of a code

Description automatically generated

## How to implement error handling in RxJS?

A screen shot of a computer code

Description automatically generated

## What does combineLatest operator work in RxJS?

CombineLatest emituje vrednost svaki put kada jedan od observable-a emituje vrednost i bitno je da je svaki od observable-a emitovao vrednost barem jednom

A computer screen with text

Description automatically generated

## difference between Promise and Observable?

A Promise handles a **single event** when an async operation completes or fails.

An Observable is like a **Stream** (in many languages) and allows you to pass zero or more events where the callback is called for each event. Primer promisa je HTTP request a observable-a click event.

## Unsubscribe in Angular - why is it important?

For this question there are two kinds of Observables - **finite** value and **infinite** value.

http Observables produce **finite** (1) values and something like a DOM event listener Observable produces **infinite** values.

If you manually call subscribe (not using async pipe), then unsubscribe from **infinite** Observables. Ukoliko se ne unsubscribe-ujemo docice do memory leak-a zato sto ce subscriber Idalje biti subscribe-ovan na observable.

## Async pipe

The async pipe is a better and more recommended way of working with observables in a component. Under the hood, the async pipe does these three tasks:

1. It subscribes to the observable and emits the last value emitted.
2. When a new value is emitted, it marks the component to be checked for the changes.
3. The async pipe automatically unsubscribes when the component is destroyed to avoid potential memory leaks.

A screenshot of a computer code

Description automatically generated

Pipe se zapocinje operatorom “ | ”

## **Observables vs Subjects**

Subject je tip observable-a koji implementira Observable i observer interfejs.

**Observables su cold sto znaci da se nece izvrsiti sve dok ne poseduje bar jednog subsrciber-a.**

**Subject je hot i on se izvrsava i kad nema subscribera.**

**Observable se izvrsava za svakog subscriber-a pojedinacno dok se kod subject-a jedna vrednost emituje svima.**

**Observable je** Uni-directional odnosno mozemo samo da primamo podatke od istog dok je Subject multidirekcioni i moze i da saljemo podatke.

A black background with white text

Description automatically generated

# How to create an Angular animation?

Angular animacije se koriste samo kada radimo sa logikom u ts ili js. Razlog je taj sto css animacije rade lakse i brze.

# How as keyword works in Angular?

currentUser$ je observable/stream i kljucna rec as nam omogucava da rezultat observable-a sacuvamo kao currentUser i time

A screenshot of a computer code

Description automatically generated

# Why is it bad to call a function in the Angular template?

A black background with white text

Description automatically generated

Kada angular vrsi change detection on proverava da li su se podaci promenili u odnosu na DOM. Zbog toga ce angular pokretati funkciju getFullName tokom svakog ciklusa provere da bi proverio da li je rezultat funkcije isti ili ne.

# What is Angular interceptor?

Presrece request koji je poslat na backend i vrsi izmene nad istim. Dobar primer je header request koji ce svakom requestu dodate header sa bitnim podacima.

A screen shot of a computer code

Description automatically generated

# Guards

Mehanizam za ogranicavanje pristupa u rutama

# LazyLoading

Since Angular creates a [SPA (Single Page Application)](https://en.wikipedia.org/wiki/Single-page_application#:~:text=From%20Wikipedia%2C%20the%20free%20encyclopedia,browser%20loading%20entire%20new%20pages.), all of its components are loaded at once. This means that a lot of unnecessary libraries or modules might be loaded as well.

Lazy loading is essential for optimizing the performance and user experience of your application. Lazy loading allows your application to load only the essential components and modules needed for the initial view. Other part of component are loading if user navigate to through the application.

# Angular Forms

Postoje 2 tipova formi: template i reactive forme

Reaktivne su bolje jer nam daju vecu kontrolu nad logikom koja se nalazi u komponenti u odnosu na template formu kod koje se logika nalazi u template-u. Ovo dovodi do toga da su reaktivne forme mnogo lakse za unit testing. Reactivne forme su zasnovane na RxJS-u i koriste observables.

Template forme se mogu koristiti za male forma dok se u ostalim slucajevima trebaju koristiti reactive forme.

# Angular inject

Koristi se za ucitavanje dependencia bez upotrebe constructora.

Inject funkcija resava problem nasledjivanja tako sto nije potrebno da koristimo super za svaku podklasu.

# Control flow

Zamena za strukturne direktive sa boljim performansama.

A black screen with yellow text

Description automatically generated

# Angular defer

Koristi se za lazy loadovanje komponenata u zavisnosti od prosledjenog uslova.

# Injectable dekorator

The @Injectable decorator provided by the Angular framework allows classes to be decorated and registered with Angular's dependency injection system.

To register the service at root level, specify 'root' as the value for the providedIn property of @Injectable. Service becomes available throughout the application by registering at the root level.

A screen shot of a computer

Description automatically generated