

Angular workshop

Timen Stepišnik Perdih

junij 2022

1 Uvod

Implementacija vseh delov projekta bo na dostopnem githubu - vsak sklop na svojem branchu. Ob razvijanju funkcionalnosti sproti pišemo unit teste.

2 Prva Angular aplikacija (1 session)

2.1 Koncepti

- ustvaritev novega Angular projekta
- pregled arhitekture angular projekta
- predstavitev koncepta komponent
- dependency injection
- osnove routinga
- "spec" datoteke in unit testi
- arrow function

2.2 Implementacija konceptov

Aplikacija s štirimi komponentami: **Home**, kjer se nahajajo gumbi za navigacijo do ostalih, **Time**, ki izpisuje trenutni čas in dve prazni.

2.3 Individualne naloge

- ustvari novo komponento, do katere je možno navigirati preko gumba v eni izmed praznih komponent

3 Spoznavanje specifik Angularja (2 sessions)

3.1 Koncepti

- pipes
- services
- ngIf
- ngFor
- osnovna uporaba API

3.2 Implementacija konceptov

Uporabimo proto dostopen API, ki vrne poljuben tekst (<https://api.publicapis.org/entries> uporabimo "description" elemente) in ga na eni izmed praznih komponent sprocesiramo z uporabo "pipe" (celotno besedilo v caps) in "service" (dictionary besed). Uporaba "pollinga" za pridobitev obdelani podatkov iz service-a, ker še ne poznamo "subscribe".

Izpis najpogostejših besed v besedilu z "ngFor".

3.3 Individualne naloge

- ustvari custom pipe za obdelavo besedila in ga uporabi na naključnem tekstu (npr. odstrani vse črke 'a')
- pripravi unit test za svoj pipe
- ustvari service ki uporabi prostodostopen API. Implementiraj smislen / zanimiv pipe, ki preoblikuje podatke in jih prikaži na eni izmed komponent

3.4 Domača naloga

Ustvari nov Angular projekt. 2 komponenti - ena naj služi kot "default" stran in ima gumb, preko katerega lahko navigiramo na drugo komponento. Na drugi komponenti naj bo gumb, ki kliče funkcijo service-a, ki poračuna številko trenutne sekunde in vrne njen ostanek pri deljenju z 11. Ostanek nato komponenta prikaže.

4 Napredni Angular koncepti (1 session)

4.1 Koncepti

- subscribe
- observables
- subject
- promise

4.2 Implementacija konceptov

Popravimo implementacijo API z uporabo asinhronne kode. Uporabimo "promise", "observable" in "subject".

4.3 Individualna / Domača naloga

Spremenimo implementacijo prejšnje domače naloge: gumb odstranimo, v service-u vsake tri sekunde poračunamo ostanek trenutne sekunde pri deljenju z 11 in rezultat z uporabo "subject" skomuniciramo komponenti, ki prikazan rezultat osvežuje. Po 10 takšnih intervalih naj "subject" pokliče metodo "complete", komponenta naj od takrat naprej namesto rezultata izpiše "zaključeno".

5 Združevanje komponent in komunikacija med njimi (2 sessions)

5.1 Koncepti

- vstavljanje ene komponente v drugo
- komunikacija med parent-child komponentami
- komunikacija med nepovezanimi komponentami

5.2 Implementacija konceptov

Pripravimo sidebar aplikacije za navigacijo do komponent, vstavimo **Time** in **Text** komponenti v prazno **Parent** komponento. Pripravimo komunikacijo med temi komponentami.

Z gumbom v eni izmed komponent spreminjamo barvo sidebar-a. Parent komponenta nekaj izpiše število vseh besed v **Text** komponenti.

5.3 Individualne naloge

- modificiraj komunikacijo med komponento in sidebarom tako, da sidebar dobi sporočilo, kakšne barve mora nastaviti svoje ozadje
- naredi **Header** komponento, ki kot input sprejme ime parent komponente. "Header" komponento vključi v vse komponente, na katere je možno navigirati, izpisuje naj ime parent komponente.

5.4 Domača naloga

Na svojem projektu implementiraj preprost sidebar ki vsebuje štiri gumb: "up", "down", "left" in "right". S pritiskom na te gumbе lahko predstavljamo gumb, ki je že implementiran (za navigacijo). Naj vsak pritisk gumb prestavi za 5px v ustrezno smer. Ob inicializaciji komponente naj bo gumb na sredini (opozorilo: tu se bo mogoče ob spremembi pozicije html elementa pojavila napaka, ki se jo odpravi z ročnim poročanjem o spremembi html predloge).

Podobno naredi še z dvema child-parent komponentama, kjer child komponenta vsebuje 4 gumbе, parent komponenta pa element, ki se premika. Na parent komponenti naj bo tudi "tarča", kamor želimo premakniti html element. Ko ga uspešno premaknemo na tarčo, parent komponenta sporoči child komponenti, da je cilj dosežen, gumbi za premikanje se nato onemogočijo.

6 Vnos uporabnika (1 session)

6.1 Koncepti

- forms
- form groups
- form validation

6.2 Implementacija konceptov

Ustvarimo novo komponento kjer pripravimo formo za uporabnikov vnos. Izmislimo si validacijo in jo implementiramo. Ob vnosu se prikaže Material stil komponente (morda expansion panel), ki izpisuje vnos uporabnika.

6.3 Individualne naloge

- dodaj še eno boolean vnosno polje poimenovano "Unordered list". Če je vnos false, se vnešeni podatki prikažejo kot <p>, sicer pa se od vnešenih podatkov prikaže le "email" kot .

6.4 Domača naloga

Ustvari novo komponento, do katere je mogoče navigirati preko gumba v sidebaru. Komponenta naj omogoča vnos uporabnika, kjer je pripravljena tudi validacija - dovoljen vnos so le črke z največjo dovoljeno dolžino 15 znakov. Ob potrditvi vnosa naj se na dnu sidebara prikaže vnos.

7 RxJS (1 session)

7.1 Koncepti

- streams of data
- reactive programming
- RxJs operators: map, filter, pipe,...

7.2 Implementacija konceptov

V projektu ustvarimo novo komponento s svojim service-om, ki simulira "stream of data". V komponenti se subscribe-amo na stream in spoznamo RxJs operatorje.

7.3 Individualne naloge

- z uporabo data stream-a premikanja miške po novi komponenti implementiraj igro, kjer je potrebno miško postaviti na skrito tarčo na zaslonu (uporabi RxJs operatorje).