

1. Uvod u Angular

Sadržaj

- O Angular *framework*-u
- Podešavanje razvojnog okruženja
- Kreiranje prve aplikacije
- Struktura Angular aplikacije
- String interpolacija
- Material Design biblioteka
- Prolazak kroz listu objekata

O Angularu

- Radni okvir (engl. *framework*) i platforma za razvoj klijentskih aplikacija korišćenjem HTML-a i TypeScript-a
- Omogućava razvoj *web* aplikacija na *frontend-u*
- Koristi TypeScript koji podržava neke objektno-orijentisane koncepte
- U arhitekturi Angular aplikacija koriste se obrasci
 - MVC
 - Dependency Injection
 - Observer
 - ...

Podešavanje razvojnog okruženja

- Instalirati Git
- Instalirati Node.js
- Instalirati Microsoft Visual Studio Code
- Instalirati Angular CLI (*Command Line Interface*) iz MS Visual Studio Code terminala navođenjem komande

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

1. If you're installing something that you want to use *in* your program, using `require('whatever')`, then install it locally, at the root of your project.
2. If you're installing something that you want to use in your *shell*, on the command line or something, install it globally, so that its binaries end up in your `PATH` environment variable.

Upravljanje verzijama

- Git mora biti instaliran na računaru
- Kreirati novi projekat pomoću Angular CLI:
`ng new lekcija1`
- Automatski se kreira i inicijalizuje lokalni git repozitorijum
- Kreirati udaljeni repozitorijum na (na github.com, na primer)
- Otvoriti direktorijum projekta u Visual Studio Code i prikazati terminal
- Ući u folder projekta i iz terminala definisati udaljeni git repozitorijum “origin”;
`git remote add origin https://github.com/djordjeprzulj/lekcija1.git`
- Uraditi *push* master grane projekta na udaljeni repozitorijum
`git push -u origin master`

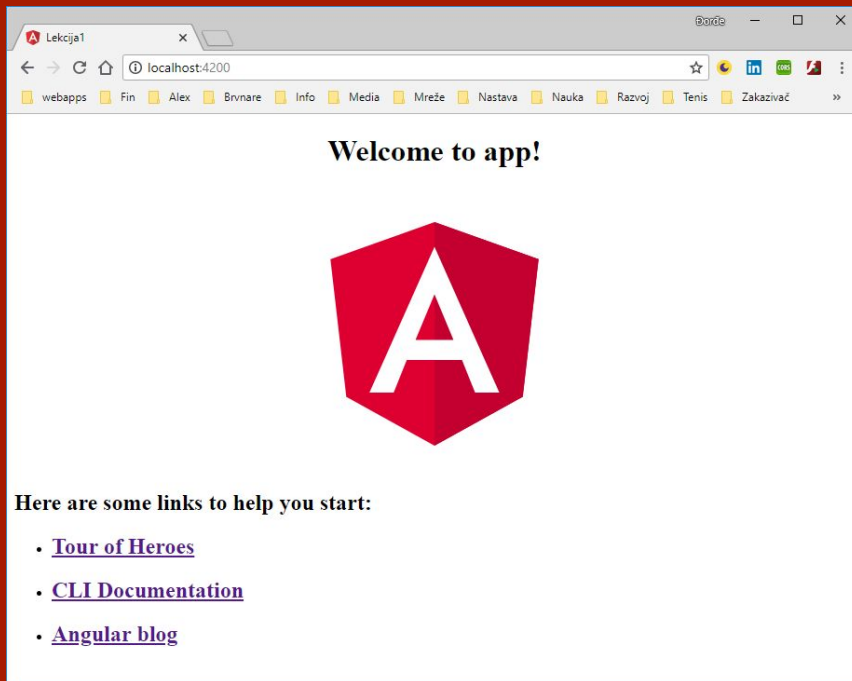
-u

--set-upstream

For every branch that is up to date or successfully pushed, add upstream (tracking) reference, used by argument-less `git-pull[1]` and other commands. For more information, see `branch.<name>.merge` in `git-config[1]`.

Pokretanje Angular aplikacije

- Iz terminala pokrenuti `ng serve`
- U browser-u pristupiti adresi `localhost:4200`
- Za prekid rada Angular aplikacije u terminalu `Ctrl+C`



Struktura Angular projekta

- `index.html`
- direktorijum `app`
 - `app.module.ts` - *root* modul
 - `app.component.*` - *root* komponenta
- `main.ts` - ulazna tačka programa
- *environments*
 - *default (development)*
 - *production*
- `node_modules` - biblioteke

```
LEKCIJA1
├─ e2e
├─ node_modules
├─ src
│  └─ app
│     ├── app.component.css
│     ├── app.component.html
│     ├── app.component.spec.ts
│     └─ app.component.ts
│
│  └─ app.module.ts
│
│  └─ assets
│
│  └─ environments
│
│  └─ browserslist
│
│  └─ favicon.ico
│
│  └─ index.html
│
│  └─ karma.conf.js
│
│  └─ main.ts
│
│  └─ polyfills.ts
│
│  └─ styles.css
│
│  └─ test.ts
```

Fajl index.html

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Lekcija1</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
<body>
  <app-root></app-root>
</body>
</html>
```


Fajl app.component.ts

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

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

Fajl app.component.html

```
<!--The content below is only a placeholder and can be replaced.-->
<div style="text-align:center">
  <h1>
    Welcome to {{ title }}!
  </h1>
  
</div>
<h2>Here are some links to help you start: </h2>
<ul>
  <li>
    <h2><a target="_blank" rel="noopener" href="https://angular.io/tutorial">Tour of Heroes</a></h2>
  </li>
  <li>...
```

Fajl app.module.ts

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';

import { AppComponent } from './app.component';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

Fajl app.main.ts

```
import { enableProdMode } from '@angular/core';
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';

import { AppModule } from './app/app.module';
import { environment } from './environments/environment';

if (environment.production) {
  enableProdMode();
}

platformBrowserDynamic().bootstrapModule(AppModule)
  .catch(err => console.log(err));
```

Ukratko o TypeScript-u

- Fajlovi sa ekstenzijom `.ts` su pisani u TypeScript jeziku
- TypeScript je nadskup JavaScript-a
 - ES5 JavaScript
 - ES6 JavaScript
 - TypeScript
- Angular je pisan u TypeScript-u
- Browser-i ne podržavaju TypeScript pa se koristi *transpiler* koji konvertuje TypeScript kod u ES6 ili ES5
 - pogledati `compilerOptions.target` u fajlu `tsconfig.json`

Nova komponenta u puzzle.component.ts

```
import { Component } from "@angular/core";
```

```
@Component({  
  selector: 'app-puzzle',  
  template: `  
    <h1>Dva lokvanja oko panja.</h1>  
    <p>Glava i usi.</p>  
  `})  
export class PuzzleComponent {  
}
```

Dodavanje PuzzleComponent u *root*

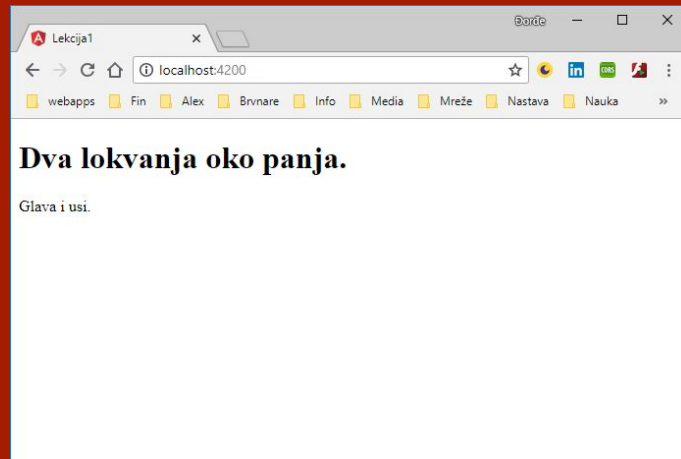
- Definisati novu klasu

```
import { Component } from "@angular/core";
```

```
@Component({  
  selector: 'app-puzzle',  
  template: `  
    <h1>Dva lokvanja oko panja.</h1>  
    <p>Glava i usi.</p>`  
})
```

```
export class PuzzleComponent {  
}
```

- U modulu navesti import i dodati u deklaracije
- U *template*-u *root* komponente navesti selektor da bi se ugradila nova komponenta



String interpolacija

- Umesto *hardcoded* navođenja u HTML, pitanje i rešenje zagonetke možemo definisati kao obeležja komponente
- Za pristup vrednostima obeležja komponente koristimo string interpolaciju

```
import { Component } from "@angular/core";
@Component({
  selector: 'app-puzzle',
  template: `
    <h1>{{ question }}</h1>
    <p>{{ solution }}</p>`
})
export class PuzzleComponent {
  question: string;
  solution: string;
  constructor() {
    this.question = 'Dva lokvanja oko panja.';
    this.solution = 'Glava i usi.';
  }
}
```


Material Design komponente za Angular

- material.angular.io
- *Framework* koji sadrži komponente za izgradnju UI
- Jedan od popularnijih, uz Twitter Bootstrap
- Veliki broj web aplikacija koristi Material Design komponente

Material biblioteke

- Instalacija Material biblioteka u Angular projekat

```
npm install --save @angular/material @angular/cdk @angular/animations
```
- Podešavanje predefinisane teme - dodati red u `styles.css`

```
@import '~@angular/material/prebuilt-themes/deeppurple-amber.css';
```
- Import modula u `AppModule` i navođenje u `imports` nizu

```
import { MatCardModule } from '@angular/material/card';  
...  
imports: [  
    BrowserModule,  
    MatCardModule  
],  
...
```

Dodavanje PuzzleListComponent

```
...  
export class PuzzleListComponent {  
  puzzles: PuzzleComponent[];  
  
  constructor() {  
    this.puzzles = [  
      {question: "Dva lokvanja oko panja.", solution: "Glava i uši."},  
      {question: "Bele koke ispod strehe vire.", solution: "Zubi."},  
      {question: "Vodu pije a živo nije.", solution: "Sunder."}  
    ];  
  }  
}
```

PuzzleListComponent *template*

```
@Component({  selector: 'app-puzzle-list',
  template: `
    <div *ngFor="let p of puzzles">
      <mat-card>
        <mat-card-header>
          <mat-card-title><h1>{{ p.question }}</h1></mat-card-title>
        </mat-card-header>
        <mat-card-content>
          <p>{{ p.solution }}</p>
        </mat-card-content>
      </mat-card>
      <br>
    </div>`
})
...
```

Direktiva `*ngFor`

- Strukturna direktiva jer menja strukturu DOM
- Izraz
`*ngFor="let p of puzzles"`
za svaki element niza kreira novi HTML element `<div>` kome je pridružen
- Pored toga, obezbeđuje promenljivu `p` preko koje se pristupa svakom elementu u nizu kroz koji se prolazi
- Sintaksa bi mogla da se čita kao
`let <naziv-koji-koristim> of <niz-obelezje-komponente>`

Rezime

- Angular je framework za razvoj *front end* aplikacija
- Koristi se HTML i TypeScript
- Angular CLI omogućava generisanje kostura projekta
- Angular aplikacija se sastoji od komponenti
- String interpolacija omogućava pristup vrednostima obeležja komponente
- Material Design biblioteka obezbeđuje kreiranje UI
- Direktiva `*ngFor` omogućava prolaz kroz niza objekata