

# Angular instalacija i osnove

Najstariji, a ujedno najmoćniji i najkompleksniji softverski okvir koji je namenjen frontend programiranju jeste Angular. U narednih nekoliko lekcija imaćete prilike da se upoznate sa osnovama korišćenja takvog softverskog okvira.

## Šta je Angular?

Angular je aplikativni softverski okvir (engl. *framework*) koji je namenjen kreiranju klijentskog dela web aplikacija. Angular je razvila kompanija Google, a prvi put je objavljen 2010. godine. To ga čini najstarijim softverskim okvirom namenjenim za frontend razvoj.



Slika 12.1. Angular – logo

Prva verzija Angulara zvala se *AngularJS* i bila je znatno drugačija od onoga što Angular danas predstavlja. S pojavom druge verzije, 2016. godine, iz naziva ovog softverskog okvira uklonjena je odrednica JS, pa Angular nastavlja da bude dostupan pod imenom koje se i danas koristi. Primena novog imena značila je i značajan zaokret na polju unutrašnje strukture i prelazak na korišćenje TypeScript jezika za unutrašnju realizaciju funkcionalnosti Angulara. Po objavljivanju verzije 2, softverski okviri Angular i AngularJS nastavili su zasebne živote. Ipak, sva pažnja razvojnog tima usmerena je na biblioteku Angular. Stoga ćemo se i mi u nastavku ovoga kursa isključivo baviti softverskim okvirom Angulara, odnosno neće biti razmatrana AngularJS varijanta.

## Uspostavljanje lokalnog razvojnog okruženja

Iako je lokalno okruženje za razvoj Angular aplikacija moguće uspostaviti samostalnim uključivanjem nekoliko osnovnih npm paketa i razvojnih alata, takav posao se u praksi obavlja korišćenjem specijalizovanog alata, baš kao što je to bio slučaj i kod React i Vue biblioteka. Mi ćemo se tokom upoznavanja Angulara pridržavati svih proverenih i preporučenih praksi, koje se koriste i tokom realnog razvoja, kako biste na najbolji način, što brže mogli da se uključite u praktičan razvoj Angular aplikacija. Stoga će uspostavljanje lokalnog razvojnog okruženja da bude obavljeno korišćenjem specijalnog alata Angular CLI.

Preduslov za korišćenje Angular CLI alata jeste instalirano izvršno okruženje Node.js i npm menadžer paketa. Dostupnost oba alata se može proveriti na već prikazane načine.

Provera dostupnosti Node.js-a:

```
node -v
```

Provera dostupnosti npm menadžera paketa:

```
npm -v
```

Obe komande treba da rezultuju prikazom podataka koji odslikavaju verzije instaliranih alata.

### Pitanje

Angular je kreirala kompanija:

- a) **Google**
- b) Facebook
- c) Microsoft
- d) Twitter

### Objašnjenje

*Angular je razvila kompanija Google, a prvi put je objavljen 2010. godine.*

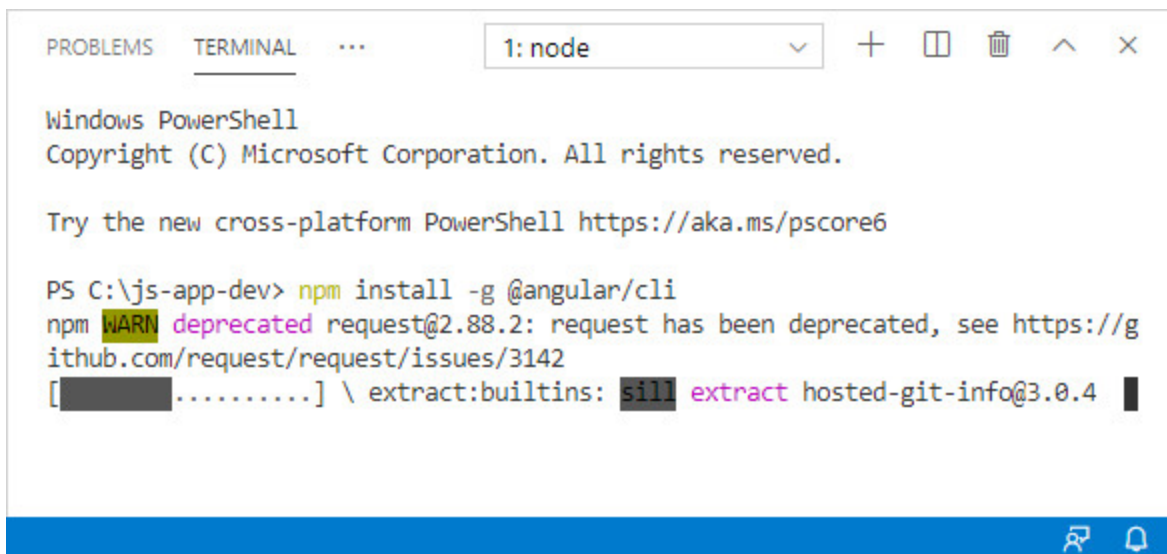
## Angular CLI

Angular CLI je alat sa komandnim interfejsom, koji se koristi za uspostavljanje razvojnog okruženja za kreiranje Angular aplikacija. Angular CLI omogućava da se na lak način izgradi kompletna projektna struktura, sa uključenim i konfigurisanim svim neophodnim razvojnim zavisnostima. Takođe, Angular CLI omogućava i pokretanje i izgradnju Angular projekata.

Angular CLI je zapravo jedan paket unutar npm repozitorijuma, koji se posredstvom izvršnog okruženja Node.js izvršava na korisničkom računaru. Stoga je njega prvo potrebno instalirati na razvojnom kompjuteru kao globalni npm paket, korišćenjem sledeće komande:

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

Na ovaj način, paket će biti instaliran globalno na razvojnom kompjuteru, zbog upotrebe oznake `-g`, prilikom upućivanja komande. Ovakvu komandu možete pokrenuti korišćenjem konzole ili terminala operativnog sistema, ili upotrebom Visual Studio Code Terminala (slika 12.2).



```
PROBLEMS  TERMINAL  ...  1: node  +  [icon]  [icon]  ^  x

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

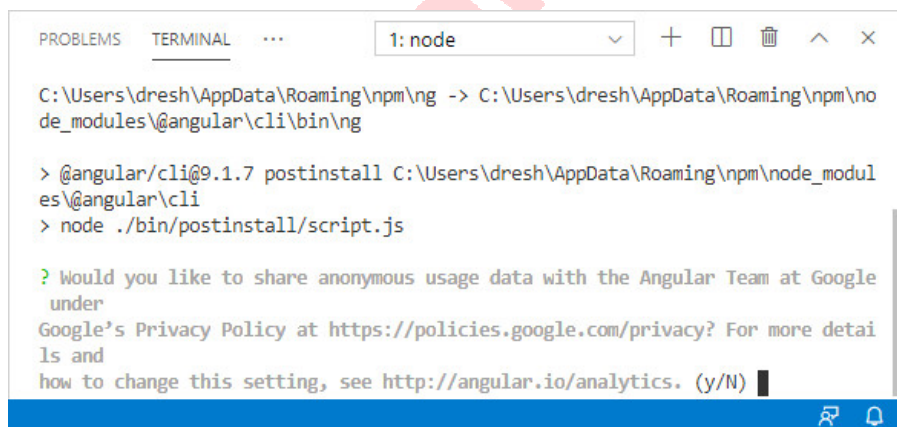
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\js-app-dev> npm install -g @angular/cli
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
[redacted] .....] \ extract:builtin: sill extract hosted-git-info@3.0.4
```

Slika 12.2. Proces instalacije alata Angular CLI

Prilikom upućivanja komande za globalno instaliranje Angular CLI alata, nije bitno na kojoj se putanji nalazite.

Na kraju instalacije može se dogoditi da dobijete pitanje u vezi sa anonimnim deljenjem podataka o korišćenju Angular proizvoda, radi njegovog budućeg unapređenja (slika 12.3).



```
PROBLEMS  TERMINAL  ...  1: node  +  [icon]  [icon]  ^  x

C:\Users\dresh\AppData\Roaming\npm\ng -> C:\Users\dresh\AppData\Roaming\npm\node_modules\@angular\cli\bin\ng

> @angular/cli@9.1.7 postinstall C:\Users\dresh\AppData\Roaming\npm\node_modules\@angular\cli
> node ./bin/postinstall/script.js

? Would you like to share anonymous usage data with the Angular Team at Google under Google's Privacy Policy at https://policies.google.com/privacy? For more details and how to change this setting, see http://angular.io/analytics. (y/N)
```

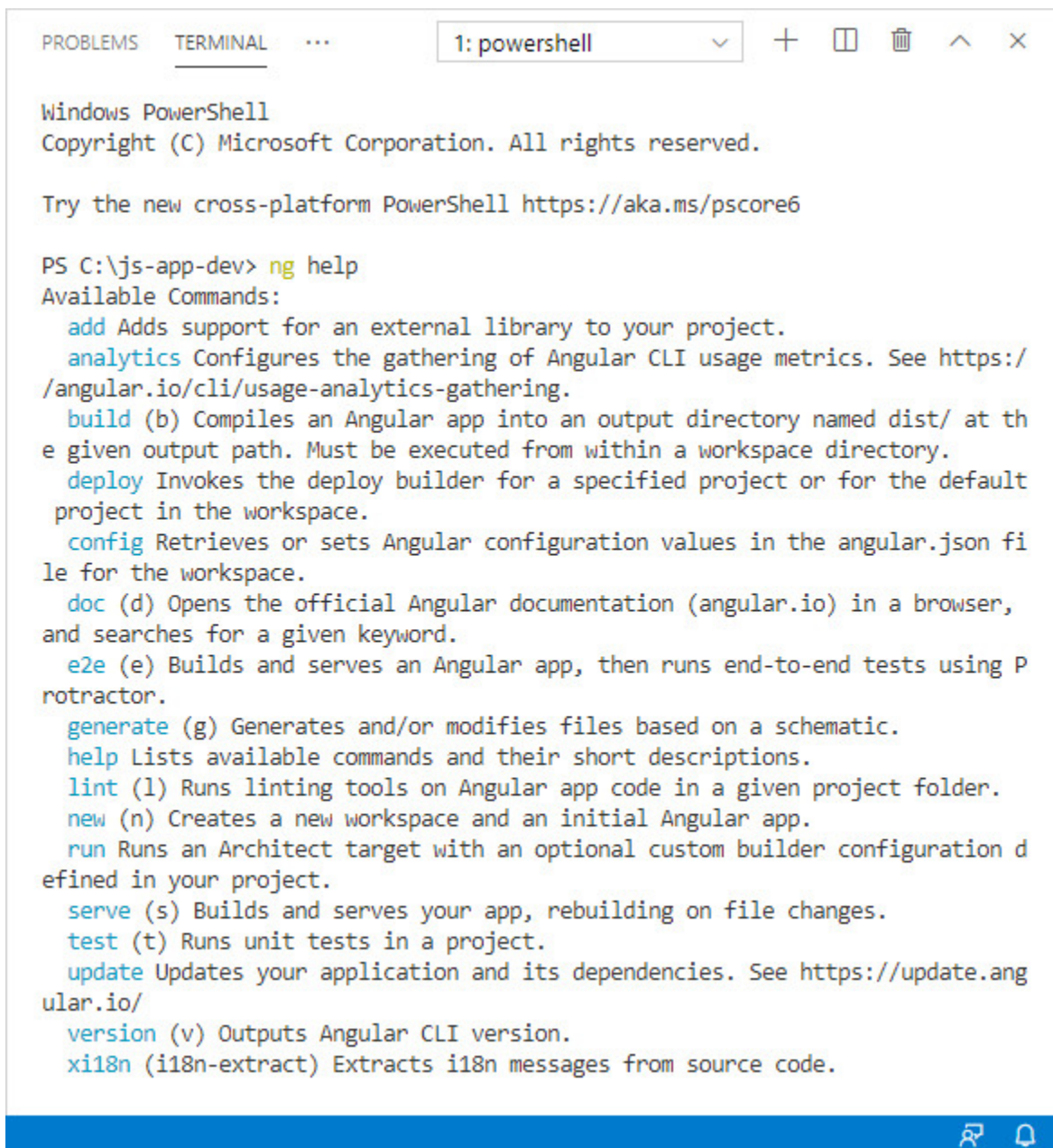
Slika 12.3. Pitanje na kraju instalacije Angular CLI

U zavisnosti od toga da li želite da omogućite anonimno deljenje podataka u vezi sa korišćenjem Angulara, potrebno je da unesete karakter N ili Y i da unos potvrdite pritiskom na taster Enter. Naravno, N se odnosi na No, a Y na Yes.

Uspešnost instalacije Angular CLI-ja možete proveriti upućivanjem komande kojom se dobija spisak svih komandi koje Angular CLI poznaje:

```
ng help
```

Ukoliko je Angular CLI uspešno instaliran, izvršavanje prikazane komande će rezultirati prikazom kao na slici 12.4.



```
PROBLEMS  TERMINAL  ...  1: powershell  +  [icon]  [icon]  ^  x

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\js-app-dev> ng help
Available Commands:
  add Adds support for an external library to your project.
  analytics Configures the gathering of Angular CLI usage metrics. See https://angular.io/cli/usage-analytics-gathering.
  build (b) Compiles an Angular app into an output directory named dist/ at the given output path. Must be executed from within a workspace directory.
  deploy Invokes the deploy builder for a specified project or for the default project in the workspace.
  config Retrieves or sets Angular configuration values in the angular.json file for the workspace.
  doc (d) Opens the official Angular documentation (angular.io) in a browser, and searches for a given keyword.
  e2e (e) Builds and serves an Angular app, then runs end-to-end tests using Protractor.
  generate (g) Generates and/or modifies files based on a schematic.
  help Lists available commands and their short descriptions.
  lint (l) Runs linting tools on Angular app code in a given project folder.
  new (n) Creates a new workspace and an initial Angular app.
  run Runs an Architect target with an optional custom builder configuration defined in your project.
  serve (s) Builds and serves your app, rebuilding on file changes.
  test (t) Runs unit tests in a project.
  update Updates your application and its dependencies. See https://update.angular.io/
  version (v) Outputs Angular CLI version.
  xi18n (i18n-extract) Extracts i18n messages from source code.
```

Slika 12.4. Spisak svih Angular CLI komandi koji se dobija komandom `ng help`

## ng

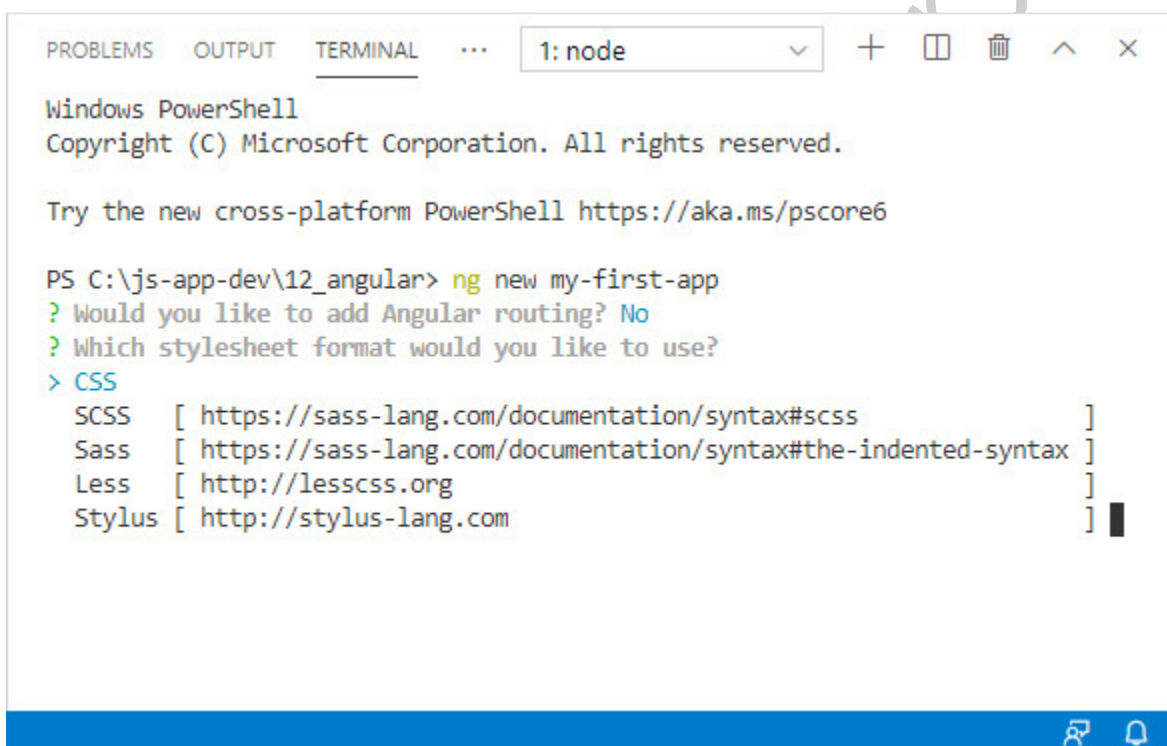
Nakon instalacije, Angular CLI je unutar konzole ili terminala dostupan uz korišćenje naziva `ng`. `ng` je zapravo skraćenica za **Angular**, a koristi se i kao prefiks prilikom imenovanja brojnih drugih elemenata Angulara.

## Kreiranje Angular radnog prostora i aplikacije

Angular aplikacije kreiraju se u kontekstu radnih prostora (engl. *workspace*). Tako se pojam radnog prostora koristi da objedini više pojedinačnih projekata u jednu zaokruženu celinu. Na taj način radni prostori omogućavaju da se kompletna web aplikacija podeli u veći broj manjih, izolovanih projekata.

Komanda za kreiranje novog Angular radnog prostora sa podrazumevanim projektom izgleda ovako:

```
ng new my-first-app
```



```
PROBLEMS OUTPUT TERMINAL ... 1: node
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\js-app-dev\12_angular> ng new my-first-app
? Would you like to add Angular routing? No
? Which stylesheet format would you like to use?
> CSS
SCSS [ https://sass-lang.com/documentation/syntax#scss ]
Sass [ https://sass-lang.com/documentation/syntax#the-indented-syntax ]
Less [ http://lesscss.org ]
Stylus [ http://stylus-lang.com ]
```

Slika 12.5. Kreiranje nove Angular aplikacije

Nakon upućivanja komande `ng new`, Angular CLI traži i određene informacije, koje će uticati na osobine projekta koji će biti generisan. To možete videti na slici 12.5. Prvo pitanje koje se dobija prilikom kreiranja nove aplikacije tiče se uključivanja funkcionalnosti za rutiranje. Drugo pitanje se odnosi na jezik koji će se unutar projekta koristiti za definisanje stilizacije. Podrazumevane postavke podrazumevaju aplikaciju bez funkcionalnosti za rutiranje, čija se stilizacija kreira korišćenjem CSS jezika. Stoga je takve podrazumevane postavke dovoljno potvrditi tasterom Enter. Naravno, ukoliko želite, možete učiniti i drugačije i tada će Angular CLI unutar razvojnog okruženja da uvrsti i konfiguriše odabrane funkcionalnosti.

Kada upotrebimo komandu `ng new`, Angular CLI kreira radni prostor i unutar njega smešta jedan projekat koji predstavlja Angular aplikaciju.

## Radni prostor i projekti Angulara

Angular aplikacije se razvijaju u kontekstu radnih prostora (engl. *workspace*). Unutar radnog prostora može se naći jedan ili više zasebnih projekata Angulara. Angular poznaje dve vrste projekata:

- projekte koji predstavljaju aplikacije i
- projekte koji predstavljaju biblioteke.

Projekat može predstavljati izvršnu, samostalnu Angular aplikaciju, ali i biblioteku koja se ne može samostalno isporučiti. Biblioteke su zamišljene kao komponente koje se mogu koristiti za modularni razvoj Angular aplikacija.

Za jednostavnije modele razvoja, jedan projekat unutar radnog prostora je sasvim dovoljan. Tek prilikom razvoja kompleksnijih aplikacija korišćenje većeg broja nezavisnih projekata može imati smisla.

Komanda `ng new` posebno je optimizovana za kreiranje radnih prostora unutar kojih će postojati samo jedan projekat, odnosno jedna Angular aplikacija. Ukoliko želite da kreirate radni prostor sa većim brojem projekata, preporučeni razvojni proces se sastoji iz prethodnog kreiranja praznog radnog prostora, unutar koga se svaki projekat dodaje pojedinačno.

Prazan radni prostor je moguće kreirati korišćenjem sledeće komande:

```
ng new my-workspace --createApplication=false
```

Ovakva komanda kreiraće radni prostor sa svim neophodnim konfiguracionim fajlovima, ali bez podrazumevanog projekta. Nakon kreiranja radnog prostora, njemu se projekti mogu dodati korišćenjem dve različite komande, u zavisnosti od tipa projekta koji želite da kreirate.

Projekat koji će predstavljati aplikaciju kreira se na sledeći način:

```
ng generate application my-first-app
```

Komanda za kreiranje Angular biblioteke izgleda ovako:

```
ng generate library my-first-library
```

Mi ćemo u nastavku ovoga kursa koristiti razvojni model koji podrazumeva postojanje jednog Angular projekta unutar radnog okruženja, što je situacija koja se dobija već prikazanom komandom `ng new`.

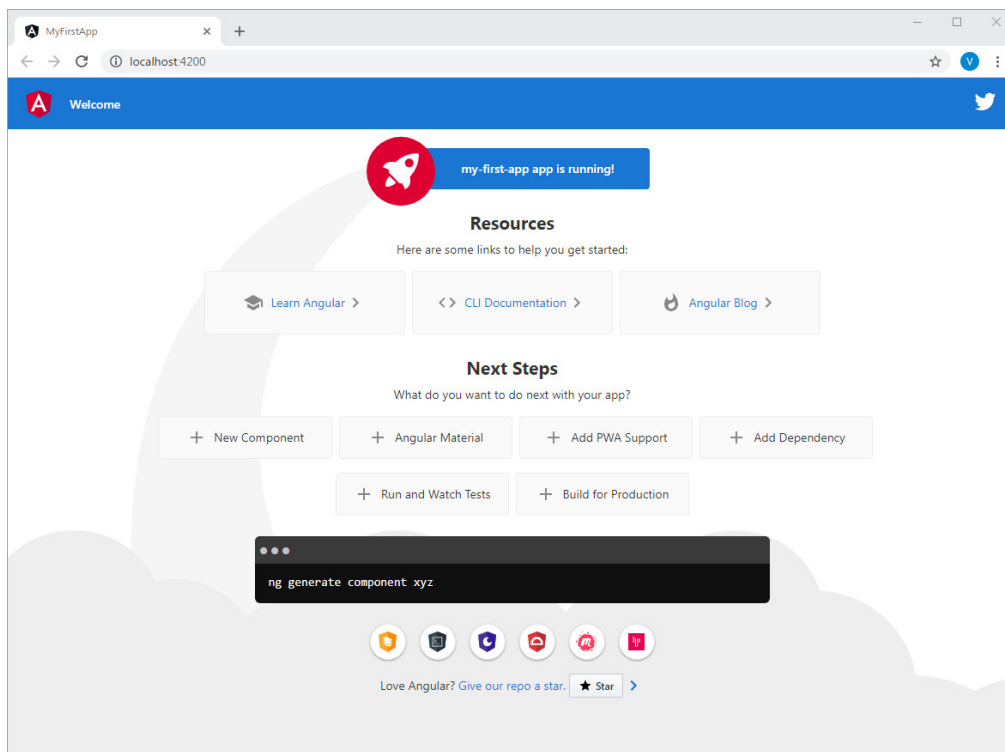
## Pokretanje aplikacije

Angular CLI u svom sastavu poseduje i razvojni lokalni server, koji se izvršava posredstvom izvršnog okruženja Node.js. Takav server omogućava da se Angular aplikacija pokrene bez prethodne izgradnje i da se njeno funkcionisanje testira. Komanda kojom se aplikacija pokreće na razvojnom serveru je sledeća:

```
ng serve --open
```



Prilikom upućivanja ovakve komande korišćenjem konzole ili terminala neophodno je biti pozicioniran unutar foldera koji predstavlja radno okruženje (engl. *workspace*). Kao što vidite, komanda `serve` je praćena dodatnom opcijom `--open`, koja će inicirati automatsko otvaranje aplikacije unutar web pregledača (slika 12.6).



*Slika 12.6. Automatski generisana prezentacija naše prve Angular aplikacije pokrenute na lokalnom serveru*

Angular aplikacija koja se hostuje na lokalnom razvojnom serveru podrazumevano je dostupna preko sledeće URL adrese:

<http://localhost:4200/>

## Struktura Angular radnog prostora i projekta

Nakon uobičajenih osnovnih operacija koje se mogu obaviti korišćenjem Angular CLI-ja, upoznaćemo se sa strukturom koja se dobija izgradnjom projekta i sa ulogom različitih fajlova i foldera koji čine takvu strukturu. Već je rečeno da komanda `ng new` obavlja kreiranje radnog prostora, unutar koga podrazumevano smešta jedan projekat koji predstavlja Angular aplikaciju.

Upućivanjem komande `ng new` obavlja se kreiranje foldera sa nazivom koji je definisan prilikom upućivanja takve komande. U našem primeru takav folder nosi naziv `my-first-app`. Svako Angular radno okruženje karakteriše se odgovarajućom fajl strukturom i postojanjem nekoliko fajlova koji se koriste za konfigurisanje projekta (slika 12.7).



Slika 12.7. Struktura Angular CLI radnog prostora

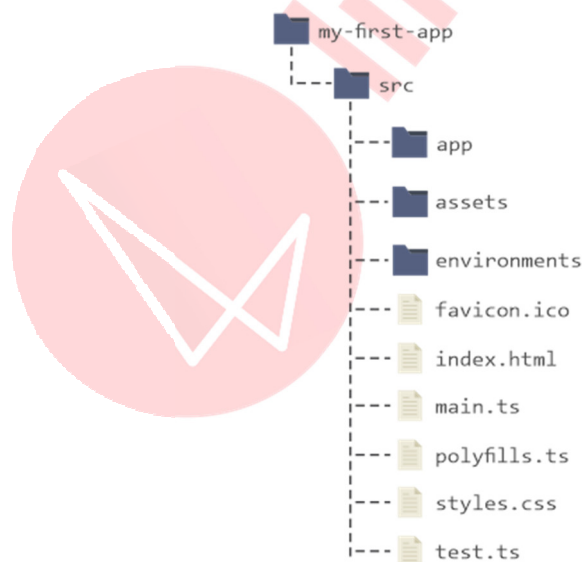
Uloga različitih fajlova i foldera unutar radnog prostora je sledeća:

- /e2e – folder sa fajlovima koji poseduju logiku za takozvano *end-to-end* testiranje (E2E, skraćeno); reč je zapravo o integralnim testovima kojima se može proveriti funkcionisanje aplikacije kao celine; za obavljanje takvih testova Angular CLI koristi biblioteku koja se zove *Protractor*,
- /node\_modules – npm paketi koje prilikom razvoja koriste svi projekti koji postoje unutar jednog radnog okruženja,
- /src – izvorni fajlovi Angular projekta koji se automatski kreira komandom `ng new`
- .editorconfig – fajl sa podešavanjima vezanim za pravila pisanja koda; podešavanja unutar ovoga fajla odnose se na konvencije koje se koriste prilikom pisanja koda, kao što su način formatiranja, odnosno uvlačenja koda, tip navodnika koji se koriste itd.; na osnovu sadržaja ovoga fajla, različiti tekst editori mogu da uspostave pravila za formatiranje koda, čime se osigurava identičan izgled koda u različitim editorima, što je prilično korisno prilikom timskog rada,
- .gitignore – fajl unutar koga se navode putanje koje je potrebno ignorisati (engl. *git ignore*) prilikom kreiranja git repozitorijuma,
- angular.json – fajl sa Angular CLI podešavanjima koja se odnose na sve projekte unutar radnog prostora; reč je o podešavanjima kojima je moguće uticati na alate za izgradnju, objavljivanje i testiranje,
- browserlist – fajl čiji sadržaj definiše web pregledače na kojima je potrebno da naša Angular aplikacija nesmetano funkcioniše; uslove za odabir podržanih web pregledača koriste Angular CLI sistemi, za dodatno konfigurisanje CSS i JavaScript koda; na osnovu sadržaja ovoga fajla, podrazumevano, Angular CLI isporučuje aplikacije koje mogu da funkcionišu na svim web pregledačima sa tržišnim učešćem većim od 0,5%; IE 9-11 nisu podržani; podešavanja iz ovoga fajla odnose se samo na jedan projekat unutar radnog prostora,



- `karma.conf.js` – fajl za podešavanje jednog posebnog alata za testiranje; reč je o alatu koji se zove *Karma*, a omogućava lako izvršavanje testova, pokretanjem test koda na specijalno kreiranom test serveru; Karma je alat koji je razvio tim Angular; podešavanja se odnose samo na jedan projekat,
- `package-lock.json` – fajl sa informacijama o verzijama svakog npm paketa koji je instaliran unutar `node_modules` foldera,
- `package.json` – fajl za podešavanje Angular aplikacije, kao jednog npm paketa,
- `README.md` – fajl unutar koga je moguće uneti osnovne tekstualne informacije o projektu,
- `tsconfig.json` – fajl za podešavanje TypeScripta, odnosno za podešavanje procesa njegovog pretvaranja u JavaScript kod; unutar ovoga foldera se definišu TypeScript podešavanja koja se primenjuju nad svim projektima unutar jednog radnog prostora; TypeScript podešavanja su podeljena u tri zasebna fajla,
- `tsconfig.app.json` – fajl sa TypeScript podešavanjima koja se primenjuju samo za jedan projekat,
- `tsconfig.spec.json` – još jedan fajl sa TypeScript podešavanjima, koja se primenjuju prilikom izgradnje testova; takođe se primenjuju samo nad projektom, a ne nad kompletnim modulom,
- `tslint.json` – fajl sa podešavanjima vezanim za alat TSLint; podešavanja se odnose na projekat, a ne na kompletan radni prostor.

Već je rečeno da se prilikom korišćenja `ng new` komande podrazumevano kreira i jedan projekat unutar koga se nalaze fajlovi Angular aplikacije. S obzirom na to da je komanda `ng new` optimizovana za kreiranje radnih okruženja sa jednim projektom, takav projekat se postavlja direktno unutar korenog foldera takvog radnog okruženja, odnosno preciznije unutar foldera **src**. Evo kako izgleda njegova struktura (slika 12.8).

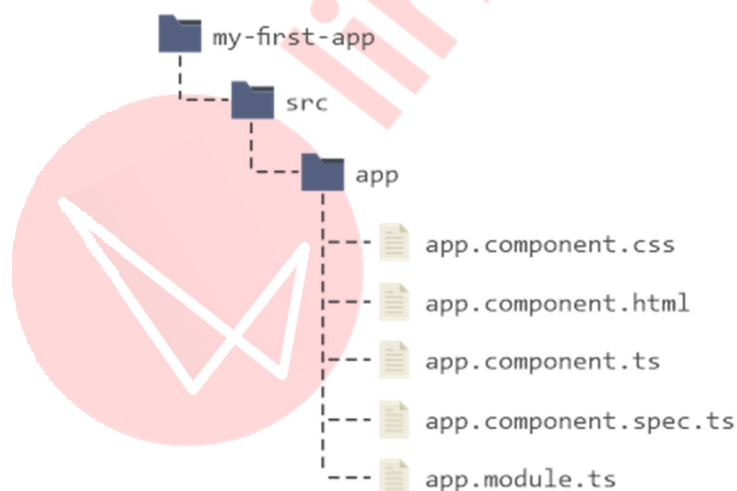


Slika 12.8. Struktura Angular CLI projekta

Uloga različitih foldera i fajlova sa izvornim kodom je sledeća:

- `app` – folder unutar koga se nalaze komponente koje predstavljaju prezentaciju i logiku naše Angular aplikacije,
- `assets` – folder za smeštanje resursa, kao što su slike, ikonice, vektori i slično; sadržaj ovog foldera se bez ikakvih transformacija uključuje u izgrađenu verziju projekta,
- `environments` – folder sa fajlovima za podešavanje procesa izgradnje projekta za različita okruženja; u ovom folderu su podrazumevano definisana dva okruženja u kojima će naša Angular aplikacija moći da se izvršava; reč je o razvojnom i produkcijskom okruženju i, za većinu scenarija, takva dva okruženja su sasvim dovoljna,
- `favicon.ico` – favicon (ikonica) web sajta,
- `index.html` – glavni HTML dokument aplikacije, unutar koga se umeće sadržaj koji se dinamički dobija izgradnjom projekta,
- `main.ts` – osnovni fajl sa logikom koja obavlja povezivanje kompletnog projekta, njegovu izgradnju i postavljanje izgrađenog koda unutar `index.html` fajla,
- `polyfills.ts` – fajl sa polyfill skriptama,
- `styles.css` – fajl za definisanje globalne stilizacije aplikacije i za uključivanje ostalih CSS fajlova kojima se stilizuju delovi aplikacije,
- `test.ts` – glavni fajl kojim se objedinjuju svi testovi.

Nešto ranije, na slici 12.6. mogli ste da vidite kako izgleda naša prva Angular aplikacija koja se automatski generiše prilikom kreiranja projekta. Ono što ste mogli da vidite unutar web pregledača direktan je produkt fajlova koji se nalaze unutar foldera **app** (slika 12.9).



Slika 12.9. Struktura fajlova koji učestvuju u izgradnji prikaza naše prve Angular aplikacije

Uloga fajlova koji se nalaze unutar foldera `app` je sledeća:

- `app.component.ts` – logika korene aplikativne komponente koja se zove `AppComponent`; i Angular se zasniva na kompoziciji komponenata, pa je tako ovo glavna komponenta unutar koje se uključuju eventualne ostale komponente,
- `app.component.css` – fajl unutar koga se definiše stilizacija glavne komponente,
- `app.component.html` – fajl unutar koga se definiše HTML šablon glavne komponente,
- `app.component.spec.ts` – fajl unutar koga se definišu testovi koji se odnose na `AppComponent` komponentu,
- `app.module.ts` – fajl koji predstavlja koreni modul aplikacije; unutar takvog modula deklariraju se svi oni elementi koji se koriste tokom razvoja; na taj način se Angularu govori na koji način je potrebno da obavi izgradnju aplikacije.

### Kreiranje Angular aplikacije sa minimalnom količinom zavisnosti

U dosadašnjem toku ove lekcije mogli ste da vidite da je struktura projekta koja se dobija korišćenjem Angular CLI-ja vrlo složena i da podrazumeva mnoštvo segmenata koji su namenjeni razvoju, objavljivanju i testiranju. Za nekoga ko tek započinje upoznavanje Angular ekosistema, dobijena struktura i količina funkcionalnosti i alata može predstavljati otežavajući faktor u razumevanju načina na koji Angular funkcioniše. Zbog toga, Angular CLI omogućava generisanje radnog prostora i projekta u uprošćenom obliku, upućivanjem sledeće komande:

```
ng new my-first-app --minimal=true
```

Dodatak na već viđenu komandu je opcija `--minimal=true`. Na ovaj način će biti kreiran radni prostor i projekat bez funkcionalnosti za testiranje. Takođe, komponente će biti kreirane u nešto kompaktnijem obliku, koji podrazumeva korišćenje samo jednog `.ts` fajla, unutar koga se objedinjuje šablon, logika i stilizacija.

Mi ćemo nastaviti rad na već kreiranom projektu.

### Izgradnja i objavljivanje aplikacije

Pre nego što se upustimo u upoznavanje načina na koji funkcioniše kreirani Angular projekat, upoznaćemo se sa još jednom značajnom komandom alata Angular CLI, koja omogućava izgradnju projekta. Kada se razvoj završi, projekat je potrebno izgraditi kako bi mogao da se objavi. Izgradnja Angular projekta se inicira korišćenjem sledeće komande:

```
ng build --prod
```

Upućivanjem ovakve komande kreira se poseban folder sa nazivom **dist** unutar korenog foldera radnog prostora. Sadržaj takvog foldera su fajlovi koje je direktno moguće prekopirati na produkcionni server.

I ovde važi identično pravilo kao i prilikom korišćenja biblioteka Vue i React. Projekat se izgrađuje pod pretpostavkom da će biti hostovan na korenoj putanji nekog domena. Ukoliko je potrebno specificovati preciznu lokaciju na kojoj će Angular aplikacija biti hostovana, to je moguće učiniti prilikom upućivanja komande za izgradnju, na sledeći način:

```
ng build --prod --base-href=/my-first-app/
```

Sada je procesu izgradnje dodata još jedna opcija `--base-href`. Vrednost ove opcije jeste bazna adresa koja će biti korišćena za izgradnju svih linkova unutar naše aplikacija. Na ovaj način Angular će zapravo prilikom izgradnje unutar finalnog `index.html` fajla da postavi sledeći element:

```
<base href="/my-first-app/">
```

Na ovaj način će biti omogućeno hostovanje naše Angular aplikacije unutar potfoldera glavnog domena, pa ukoliko Angular aplikaciju objavite na lokalnom Apache serveru, ona će biti dostupna preko sledeće adrese:

<http://localhost/my-first-app/>

## Kako funkcioniše upravo kreirani Angular projekat?

Način na koji funkcioniše upravo kreirana Angular aplikacija najlakše je razumeti ukoliko pratimo tok podataka, odnosno ukoliko utvrdimo odakle dolaze podaci koji su prikazani unutar web pregledača. Na slici 12.6. možete da vidite kako izgleda naša aplikacija kada se otvori unutar web pregledača. HTML dokument započinje tekstem *my-first-app app is running!* Hajde da pogledamo odakle dolazi takav tekst.

Unutar `app.component.html`, možete pronaći sledeću liniju koda:

```
<span>{{ title }} app is running!</span>
```

Ova linija zaslužna je za prikaz početne poruke koju možete videti unutar web pregledača. Ipak, možete videti i to da sadržaj prikazanog `span` elementa unutar fajla nije isti kao i poruka koja se dobija unutar pregledača. Jednostavno, umesto `{{ title }}`, unutar izgrađene stranice postavlja se vrednost promenljive `title`. Gde je takva promenljiva definisana?

Promenljiva `title` definisana je unutar fajla `app.component.ts`:

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

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

Pa čak iako ne poznajete TypeScript kojim je kod ovakvog fajla napisan, određeni delovi su razumljivi. Unutar klase `AppComponent` možete da vidite naredbu za inicijalizaciju promenljive `title`. Njena vrednost je tekst koji se na kraju prikazuje unutar web pregledača.

Ipak, iako je TypeScript prilično jednostavan za učenje, pogotovo za nekoga ko dobro poznaje JavaScript, mi ćemo kompletnu narednu lekciju posvetiti upoznavanju specifičnih osobina TypeScripta, koje takav jezik distanciraju od jezika JavaScript.

## Rezime

- Angular je aplikativni softverski okvir (engl. *framework*) koji je namenjen kreiranju klijentskog dela web aplikacija.
- Angular je razvila kompanija Google, a prvi put je obavljen 2010. godine.
- Prva verzija Angulara zvala se *AngularJS* i bila je znatno drugačija od onoga što Angular danas predstavlja.
- Angular CLI je alat sa komandnim interfejsom, koji se koristi za uspostavljanje razvojnog okruženja za kreiranje Angular aplikacija.
- Angular CLI se instalira kao globalni npm paket, korišćenjem komande `npm install -g @angular/cli`.
- Nakon instalacije, Angular CLI je unutar konzole ili terminala dostupan uz korišćenje naziva `ng`.
- `ng` je skraćenica za **Angular**.
- Komanda za kreiranje novog radnog prostora Angulara sa podrazumevanim projektom je `ng new my-first-app`, gde se `my-first-app` odnosi na naziv projekta.
- Angular aplikacije se razvijaju u kontekstu radnih prostora koji mogu biti sačinjeni iz jednog ili više projekata.
- Angular aplikaciju je moguće pokrenuti na razvojnem serveru korišćenjem komande `ng serve -open`.
- Izgradnja Angular projekta se inicira korišćenjem komande `ng build -prod`.

