**Portfolio**

**Angular.json**

{

"$schema": "./node\_modules/@angular/cli/lib/config/schema.json",

"version": 1,

"newProjectRoot": "projects",

"projects": {

"myPortfolio": {

"root": "",

"sourceRoot": "src",

"projectType": "application",

"prefix": "app",

"schematics": {},

"architect": {

"build": {

"builder": "@angular-devkit/build-angular:browser",

"options": {

"outputPath": "dist/myPortfolio",

"index": "src/index.html",

"main": "src/main.ts",

"polyfills": "src/polyfills.ts",

"tsConfig": "src/tsconfig.app.json",

"assets": [

"src/favicon.ico",

"src/assets"

],

"styles": [

"node\_modules/bootstrap/dist/css/bootstrap.min.css",

"src/styles.css"

],

"scripts": [],

"es5BrowserSupport": true

},

"configurations": {

"production": {

"fileReplacements": [

{

"replace": "src/environments/environment.ts",

"with": "src/environments/environment.prod.ts"

}

],

"optimization": true,

"outputHashing": "all",

"sourceMap": false,

"extractCss": true,

"namedChunks": false,

"aot": true,

"extractLicenses": true,

"vendorChunk": false,

"buildOptimizer": true,

"budgets": [

{

"type": "initial",

"maximumWarning": "2mb",

"maximumError": "5mb"

}

]

}

}

},

"serve": {

"builder": "@angular-devkit/build-angular:dev-server",

"options": {

"browserTarget": "myPortfolio:build"

},

"configurations": {

"production": {

"browserTarget": "myPortfolio:build:production"

}

}

},

"extract-i18n": {

"builder": "@angular-devkit/build-angular:extract-i18n",

"options": {

"browserTarget": "myPortfolio:build"

}

},

"test": {

"builder": "@angular-devkit/build-angular:karma",

"options": {

"main": "src/test.ts",

"polyfills": "src/polyfills.ts",

"tsConfig": "src/tsconfig.spec.json",

"karmaConfig": "src/karma.conf.js",

"styles": [

"src/styles.css"

],

"scripts": [],

"assets": [

"src/favicon.ico",

"src/assets"

]

}

},

"lint": {

"builder": "@angular-devkit/build-angular:tslint",

"options": {

"tsConfig": [

"src/tsconfig.app.json",

"src/tsconfig.spec.json"

],

"exclude": [

"\*\*/node\_modules/\*\*"

]

}

}

}

},

"myPortfolio-e2e": {

"root": "e2e/",

"projectType": "application",

"prefix": "",

"architect": {

"e2e": {

"builder": "@angular-devkit/build-angular:protractor",

"options": {

"protractorConfig": "e2e/protractor.conf.js",

"devServerTarget": "myPortfolio:serve"

},

"configurations": {

"production": {

"devServerTarget": "myPortfolio:serve:production"

}

}

},

"lint": {

"builder": "@angular-devkit/build-angular:tslint",

"options": {

"tsConfig": "e2e/tsconfig.e2e.json",

"exclude": [

"\*\*/node\_modules/\*\*"

]

}

}

}

}

},

"defaultProject": "myPortfolio"

}

**Package.json**

{

"name": "my-portfolio",

"version": "0.0.0",

"scripts": {

"ng": "ng",

"start": "ng serve",

"build": "ng build",

"test": "ng test",

"lint": "ng lint",

"e2e": "ng e2e"

},

"private": true,

"dependencies": {

"@angular/animations": "~7.2.0",

"@angular/common": "~7.2.0",

"@angular/compiler": "~7.2.0",

"@angular/core": "~7.2.0",

"@angular/forms": "~7.2.0",

"@angular/platform-browser": "~7.2.0",

"@angular/platform-browser-dynamic": "~7.2.0",

"@angular/router": "~7.2.0",

"bootstrap": "^4.3.1",

"core-js": "^2.5.4",

"rxjs": "~6.3.3",

"tslib": "^1.9.0",

"zone.js": "~0.8.26"

},

"devDependencies": {

"@angular-devkit/build-angular": "~0.13.0",

"@angular/cli": "~7.3.8",

"@angular/compiler-cli": "~7.2.0",

"@angular/language-service": "~7.2.0",

"@types/node": "~8.9.4",

"@types/jasmine": "~2.8.8",

"@types/jasminewd2": "~2.0.3",

"codelyzer": "~4.5.0",

"jasmine-core": "~2.99.1",

"jasmine-spec-reporter": "~4.2.1",

"karma": "~4.0.0",

"karma-chrome-launcher": "~2.2.0",

"karma-coverage-istanbul-reporter": "~2.0.1",

"karma-jasmine": "~1.1.2",

"karma-jasmine-html-reporter": "^0.2.2",

"protractor": "~5.4.0",

"ts-node": "~7.0.0",

"tslint": "~5.11.0",

"typescript": "~3.2.2"

}

}

**tslint.json**

{

"extends": "tslint:recommended",

"rulesDirectory": [

"codelyzer"

],

"rules": {

"array-type": false,

"arrow-parens": false,

"deprecation": {

"severity": "warn"

},

"import-blacklist": [

true,

"rxjs/Rx"

],

"interface-name": false,

"max-classes-per-file": false,

"max-line-length": [

true,

140

],

"member-access": false,

"member-ordering": [

true,

{

"order": [

"static-field",

"instance-field",

"static-method",

"instance-method"

]

}

],

"no-consecutive-blank-lines": false,

"no-console": [

true,

"debug",

"info",

"time",

"timeEnd",

"trace"

],

"no-empty": false,

"no-inferrable-types": [

true,

"ignore-params"

],

"no-non-null-assertion": true,

"no-redundant-jsdoc": true,

"no-switch-case-fall-through": true,

"no-use-before-declare": true,

"no-var-requires": false,

"object-literal-key-quotes": [

true,

"as-needed"

],

"object-literal-sort-keys": false,

"ordered-imports": false,

"quotemark": [

true,

"single"

],

"trailing-comma": false,

"no-output-on-prefix": true,

"use-input-property-decorator": true,

"use-output-property-decorator": true,

"use-host-property-decorator": true,

"no-input-rename": true,

"no-output-rename": true,

"use-life-cycle-interface": true,

"use-pipe-transform-interface": true,

"component-class-suffix": true,

"directive-class-suffix": true

}

}

**Protractor.config.js**

// Protractor configuration file, see link for more information

// https://github.com/angular/protractor/blob/master/lib/config.ts

const { SpecReporter } = require('jasmine-spec-reporter');

exports.config = {

allScriptsTimeout: 11000,

specs: [

'./src/\*\*/\*.e2e-spec.ts'

],

capabilities: {

'browserName': 'chrome'

},

directConnect: true,

baseUrl: 'http://localhost:4200/',

framework: 'jasmine',

jasmineNodeOpts: {

showColors: true,

defaultTimeoutInterval: 30000,

print: function() {}

},

onPrepare() {

require('ts-node').register({

project: require('path').join(\_\_dirname, './tsconfig.e2e.json')

});

jasmine.getEnv().addReporter(new SpecReporter({ spec: { displayStacktrace: true } }));

}

};

**App.ts**

import { AppPage } from './app.po';

import { browser, logging } from 'protractor';

describe('workspace-project App', () => {

let page: AppPage;

beforeEach(() => {

page = new AppPage();

});

it('should display welcome message', () => {

page.navigateTo();

expect(page.getTitleText()).toEqual('Welcome to myPortfolio!');

});

afterEach(async () => {

// Assert that there are no errors emitted from the browser

const logs = await browser.manage().logs().get(logging.Type.BROWSER);

expect(logs).not.toContain(jasmine.objectContaining({

level: logging.Level.SEVERE,

} as logging.Entry));

});

});

**App.po.ts**

import { browser, by, element } from 'protractor';

export class AppPage {

navigateTo() {

return browser.get(browser.baseUrl) as Promise<any>;

}

getTitleText() {

return element(by.css('app-root h1')).getText() as Promise<string>;

}

}

About-us.components.ts

import { Component, OnInit } from '@angular/core';

@Component({

selector: 'app-about-us',

templateUrl: './about-us.component.html',

styleUrls: ['./about-us.component.css']

})

export class AboutUsComponent implements OnInit {

constructor() { }

ngOnInit() {

}

}

App.components.css

.fill {

height: 75% !important;

width: 100%;

position: absolute;

top: 0;

left: 0;

overflow:hidden;

background-size: cover;

background-position: center center;

}

.ima{

height: 25px;

width: 25px;

}

App.components.html

<div class="container">

<nav class="navbar navbar-expand-lg navbar-light ">

<ul class="nav navbar-nav float-left" routerLinkActive="active">

<li class="nav-item"> <img class="ima" src="http://clipart-library.com/data\_images/206118.jpg" alt="image"><a class="btn" routerLink="/">myPortFolio</a></li>

</ul>

<ul class="nav navbar-nav " style="margin-left: 70%" routerLinkActive="active">

<li class="nav-item"><a class=" btn" routerLink="/signin" style="margin: 5px;">Sign In</a></li>

<li class="nav-item"><a class="btn" routerLink="/signup" style="margin: 5px">Sign Up</a> </li>

</ul>

</nav>

<div class="container">

<router-outlet></router-outlet>

</div>

</div>

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.error(err));