Index.html

<!doctype html>

<html lang="en">

<head>

  <meta charset="utf-8">

  <title>Simply Learn Test Project</title>

  <base href="/">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="icon" type="image/x-icon" href="favicon.ico">

<link

href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"

rel="stylesheet" integrity="sha384-

1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"

crossorigin="anonymous">

</head>

<body>

  <app-root></app-root>

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p" crossorigin="anonymous"></script>

</body>

</html>

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));

Test.ts

import 'zone.js/testing';

import { getTestBed } from '@angular/core/testing';

import {

  BrowserDynamicTestingModule,

  platformBrowserDynamicTesting

} from '@angular/platform-browser-dynamic/testing';

declare const require: {

  context(path: string, deep?: boolean, filter?: RegExp): {

    <T>(id: string): T;

    keys(): string[];

  };

};

// First, initialize the Angular testing environment.

getTestBed().initTestEnvironment(

  BrowserDynamicTestingModule,

  platformBrowserDynamicTesting(),

);

// Then we find all the tests.

const context = require.context('./', true, /\.spec\.ts$/);

// And load the modules.

context.keys().map(context);

angular.json

{

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

  "version": 1,

  "newProjectRoot": "projects",

  "projects": {

    "simply\_learn\_test\_project": {

      "projectType": "application",

      "schematics": {

        "@schematics/angular:application": {

          "strict": true

        }

      },

      "root": "",

      "sourceRoot": "src",

      "prefix": "app",

      "architect": {

        "build": {

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

          "options": {

            "outputPath": "dist/simply\_learn\_test\_project",

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

            "main": "src/main.ts",

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

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

            "assets": [

              "src/favicon.ico",

              "src/assets"

            ],

            "styles": [

              "src/styles.css"

            ],

            "scripts": []

          },

          "configurations": {

            "production": {

              "budgets": [

                {

                  "type": "initial",

                  "maximumWarning": "500kb",

                  "maximumError": "1mb"

                },

                {

                  "type": "anyComponentStyle",

                  "maximumWarning": "2kb",

                  "maximumError": "4kb"

                }

              ],

              "fileReplacements": [

                {

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

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

                }

              ],

              "outputHashing": "all"

            },

            "development": {

              "buildOptimizer": false,

              "optimization": false,

              "vendorChunk": true,

              "extractLicenses": false,

              "sourceMap": true,

              "namedChunks": true

            }

          },

          "defaultConfiguration": "production"

        },

        "serve": {

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

          "configurations": {

            "production": {

              "browserTarget": "simply\_learn\_test\_project:build:production"

            },

            "development": {

              "browserTarget": "simply\_learn\_test\_project:build:development"

            }

          },

          "defaultConfiguration": "development"

        },

        "extract-i18n": {

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

          "options": {

            "browserTarget": "simply\_learn\_test\_project:build"

          }

        },

        "test": {

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

          "options": {

            "main": "src/test.ts",

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

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

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

            "assets": [

              "src/favicon.ico",

              "src/assets"

            ],

            "styles": [

              "src/styles.css"

            ],

            "scripts": []

          }

        }

      }

    }

  },

  "defaultProject": "simply\_learn\_test\_project"

}

App.component.css

module.exports = function (config) {

  config.set({

    basePath: '',

    frameworks: ['jasmine', '@angular-devkit/build-angular'],

    plugins: [

      require('karma-jasmine'),

      require('karma-chrome-launcher'),

      require('karma-jasmine-html-reporter'),

      require('karma-coverage'),

      require('@angular-devkit/build-angular/plugins/karma')

    ],

    client: {

      jasmine: {

        // you can add configuration options for Jasmine here

        // the possible options are listed at https://jasmine.github.io/api/edge/Configuration.html

        // for example, you can disable the random execution with `random: false`

        // or set a specific seed with `seed: 4321`

      },

      clearContext: false // leave Jasmine Spec Runner output visible in browser

    },

    jasmineHtmlReporter: {

      suppressAll: true // removes the duplicated traces

    },

    coverageReporter: {

      dir: require('path').join(\_\_dirname, './coverage/simply\_learn\_test\_project'),

      subdir: '.',

      reporters: [

        { type: 'html' },

        { type: 'text-summary' }

      ]

    },

    reporters: ['progress', 'kjhtml'],

    port: 9876,

    colors: true,

    logLevel: config.LOG\_INFO,

    autoWatch: true,

    browsers: ['Chrome'],

    singleRun: false,

    restartOnFileChange: true

  });

};

App-routing.module.ts

import { NgModule } from '@angular/core';

import { RouterModule, Routes } from '@angular/router';

import { AboutComponent } from './components/about/about.component';

import { HomeComponent } from './components/home/home.component';

import { ResultComponent } from './components/result/result.component';

//declare all routes below

const routes: Routes = [

  {path:'',redirectTo:'home',pathMatch:'full'},

{path:'home',component:HomeComponent},

{path:'about',component:AboutComponent},

{path:'result',component:ResultComponent}

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

app.component.spec.ts

import { TestBed } from '@angular/core/testing';

import { RouterTestingModule } from '@angular/router/testing';

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

describe('AppComponent', () => {

  beforeEach(async () => {

    await TestBed.configureTestingModule({

      imports: [

        RouterTestingModule

      ],

      declarations: [

        AppComponent

      ],

    }).compileComponents();

  });

  it('should create the app', () => {

    const fixture = TestBed.createComponent(AppComponent);

    const app = fixture.componentInstance;

    expect(app).toBeTruthy();

  });

  it(`should have as title 'simply\_learn\_test\_project'`, () => {

    const fixture = TestBed.createComponent(AppComponent);

    const app = fixture.componentInstance;

    expect(app.title).toEqual('simply\_learn\_test\_project');

  });

  it('should render title', () => {

    const fixture = TestBed.createComponent(AppComponent);

    fixture.detectChanges();

    const compiled = fixture.nativeElement as HTMLElement;

    expect(compiled.querySelector('.content span')?.textContent).toContain('simply\_learn\_test\_project app is running!');

  });

});

app.component.ts

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

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  title = 'simply\_learn\_test\_project';

}

app.module.ts

import { NgModule } from '@angular/core';

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

import { AppRoutingModule } from './app-routing.module';

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

import { HeaderComponent } from './components/header/header.component';

import { HomeComponent } from './components/home/home.component';

import { AboutComponent } from './components/about/about.component';

import {HttpClientModule} from '@angular/common/http'

import { FormsModule } from '@angular/forms';

import { ResultComponent } from './components/result/result.component';

@NgModule({

  declarations: [

    AppComponent,

    HeaderComponent,

    HomeComponent,

    AboutComponent,

    ResultComponent

      ],

  imports: [

    BrowserModule,

    AppRoutingModule,

    HttpClientModule,

    FormsModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }