jualoppaz / **single-spa-angular-app**

Angular v8 application with two example pages for be included in a single-spa application as registered app.

MIT License

2 stars 4 forks

Star Watch

Code Issues Pull requests 8 Actions Projects Wiki Security

jualoppaz Update README.md ... on Oct 27, 2020 **90**

View code

README.md

npm v0.1.5

single-spa-angular-app

Angular v8 application with two example pages for be included in a single-spa application as registered app. This is an Angular v8 application example used as NPM package in single-spa-login-example-with-npm-packages in order to register an application. See the installation instructions there.

□□ Motivation

This is an Angular v8 application which contains two routed pages for embbed the app inside a root single-spa application.

How it works ❓

There are several files for the right working of this application and they are:

- src/app/app.module.ts
- src/app/app-routing.module.ts
- src/main.single-spa.ts
- angular.json
- extra-webpack.config.ts
- package.json

This file has no custom config. But we must set desired config here if needed.

src/app/app.module.ts

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

import {APP_BASE_HREF} from '@angular/common';
import { ListComponent } from './list/list.component';
    BrowserModule,
    AppRoutingModule,
  ],
  providers: [
    {provide: APP_BASE_HREF, useValue: '/'}
  ],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

As this application will be mounted when browser url starts with **/angular**, we need to **provide** the **APP_BASE_HREF** property with **/angular** value. However, as is documented here, this config causes strange behaviours in angular router when navigating between registered apps.

A simple way of avoid this is set **/** as value of **APP_BASE_HREF** property and repeat **angular** prefix in all routes as you can see in **app-routing.module.ts** file.

src/app/app-routing.module.ts

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

import {ListComponent} from './list/list.component';
import {DetailComponent} from './detail/detail.component';
import {EmptyRouteComponent} from './empty-route/empty-route.component';

const routes: Routes = [
```

```
  { path: 'angular', component: ListComponent },
  { path: 'angular/detail',      component: DetailComponent },
  { path: '**', component: EmptyRouteComponent }
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

As it is explained in **src/app/app.module.ts** section we need to add **angular** prefix in every routes.

src/main.single-spa.ts

```
import { enableProdMode, NgZone } from '@angular/core';

import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
import { Router } from '@angular/router';
import { AppModule } from './app/app.module';
import { environment } from './environments/environment';
import singleSpaAngular from 'single-spa-angular';
import { singleSpaPropsSubject } from './single-spa/single-spa-props';

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

const lifecycles = singleSpaAngular({
  bootstrapFunction: singleSpaProps => {
    singleSpaPropsSubject.next(singleSpaProps);
    return platformBrowserDynamic().bootstrapModule(AppModule);
  },
  template: '<app-root />',
  Router,
  NgZone,
  domElementGetter: () => document.getElementById('angular-app')
});

export const bootstrap = lifecycles.bootstrap;
export const mount = lifecycles.mount;
export const unmount = lifecycles.unmount;
```

The **lifecycles** object contains all **single-spa-angular** methods for the **single-spa** lifecycle of this app. All used config is default one but the custom config of the **domElementGetter** option. It's assumed that an element with **angular-app** id is defined in the **index.html** where this application will be mounted.

angular.json

```
{
  "$schema": "./node_modules/@angular/cli/lib/config/schema.json",
  "version": 1,
  "newProjectRoot": "projects",
  "projects": {
    "single-spa-angular-app": {
      "projectType": "application",
      "schematics": {
        "@schematics/angular:component": {
          "style": "scss"
        }
      },
      "root": "",
      "sourceRoot": "src",
      "prefix": "app",
      "architect": {
        "build": {
          "builder": "@angular-builders/custom-webpack:browser",
          "options": {
            "outputPath": "dist",
            "index": "src/index.html",
            "main": "src/main.single-spa.ts",
            "polyfills": "src/polyfills.ts",
            "tsConfig": "tsconfig.app.json",
            "aot": false,
            "assets": [
              "src/favicon.ico",
              "src/assets"
            ],
            "styles": [
              "src/styles.scss"
            ],
            "scripts": [],
            "customWebpackConfig": {
              "path": "./extra-webpack.config.js"
            }
          },
          "configurations": {
            "production": {
              "fileReplacements": [
                {
                  "replace": "src/environments/environment.ts",
```

```
                "with": "src/environments/environment.prod.ts"
              }
            ],
            "optimization": true,
            "outputHashing": "none",
            "sourceMap": false,
            "extractCss": true,
            "namedChunks": false,
            "aot": true,
            "extractLicenses": true,
            "vendorChunk": false,
            "buildOptimizer": true,
            "budgets": [
              {
                "type": "initial",
                "maximumWarning": "2mb",
                "maximumError": "5mb"
              },
              {
                "type": "anyComponentStyle",
                "maximumWarning": "6kb",
                "maximumError": "10kb"
              }
            ]
          }
        }
      },
      "serve": {
        "builder": "@angular-builders/custom-webpack:dev-server",
        "options": {
          "browserTarget": "single-spa-angular-app:build"
        },
        "configurations": {
          "production": {
            "browserTarget": "single-spa-angular-app:build:production"
          }
        }
      },
      "extract-i18n": {
        "builder": "@angular-devkit/build-angular:extract-i18n",
        "options": {
          "browserTarget": "single-spa-angular-app: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.scss"
            ],
            "scripts": []
          }
        },
        "lint": {
          "builder": "@angular-devkit/build-angular:tslint",
          "options": {
            "tsConfig": [
              "tsconfig.app.json",
              "tsconfig.spec.json",
              "e2e/tsconfig.json"
            ],
            "exclude": [
              "**/node_modules/**"
            ]
          }
        },
        "e2e": {
          "builder": "@angular-devkit/build-angular:protractor",
          "options": {
            "protractorConfig": "e2e/protractor.conf.js",
            "devServerTarget": "single-spa-angular-app:serve"
          },
          "configurations": {
            "production": {
              "devServerTarget": "single-spa-angular-app:serve:production"
            }
          }
        }
      }
    },
  },
  "defaultProject": "single-spa-angular-app"
}
```

The essential config is in **@angular-builders/custom-webpack:browser** builder. Be care of this config is autogenerated when install **single-spa-angular**.

extra-webpack.config.js

```
const singleSpaAngularWebpack = require('single-spa-angular/lib/webpack').default

module.exports = (angularWebpackConfig, options) => {
  const singleSpaWebpackConfig = singleSpaAngularWebpack(angularWebpackConfig, options)
```

```
  // Feel free to modify this webpack config however you'd like to
  return singleSpaWebpackConfig
}
```

package.json

```
{
  "name": "single-spa-angular-app",
  "version": "0.1.5",
  "description": "Angular v8 application with two example pages for be included in a sing
  "main": "dist/main-es2015.js",
  "scripts": {
    "ng": "ng",
    "test": "ng test",
    "lint": "ng lint",
    "build:single-spa": "ng build --prod"
  },
  "dependencies": {
    "@angular-builders/custom-webpack": "8.4.1",
    "@angular/common": "8.2.14",
    "@angular/compiler": "8.2.14",
    "@angular/core": "8.2.14",
    "@angular/forms": "8.2.14",
    "@angular/platform-browser": "8.2.14",
    "@angular/platform-browser-dynamic": "8.2.14",
    "@angular/router": "8.2.14",
    "rxjs": "6.4.0",
    "single-spa-angular": "3.1.0",
    "tslib": "1.10.0",
    "zone.js": "0.9.1"
  },
  "devDependencies": {
    "@angular-devkit/build-angular": "0.803.23",
    "@angular-devkit/build-ng-packagr": "0.803.23",
    "@angular/cli": "8.3.23",
    "@angular/compiler-cli": "8.2.14",
    "@angular/language-service": "8.2.14",
    "@types/node": "8.9.5",
    "@types/jasmine": "3.3.16",
    "@types/jasminewd2": "2.0.8",
    "codelyzer": "5.2.1",
    "jasmine-core": "3.4.0",
    "jasmine-spec-reporter": "4.2.1",
    "karma": "4.1.0",
    "karma-chrome-launcher": "2.2.0",
    "karma-coverage-istanbul-reporter": "2.0.6",
    "karma-jasmine": "2.0.1",
    "karma-jasmine-html-reporter": "1.5.1",
    "ng-packagr": "5.7.1",
```

```
    "protractor": "5.4.2",
    "ts-node": "7.0.1",
    "tsickle": "0.37.1",
    "tslint": "5.15.0",
    "typescript": "3.5.3"
  }
}
```

There are several scripts in this project:

- **ng**: for use global ng cli
- **test**: use for run unit tests
- **lint**: for run **eslint** in all project
- **build:single-spa**: for compile the application and build it as a **libray** in **umd** format