Angular Client Installation Guide

Version 1.0

Published: 14/11/19

VERBAT

Who should read this document:

This document is *ideally* meant for IT Staff as it is technical in nature

a. System Administrators
b. DevOps team members
c. Network experts
d. Project administrator's seeking technical guidance for installation of the 'OER' project.

Pre-requisites:

1. Install CentOS Server on an server class computer in a location of your choice
2. Make sure you set up remote access complying to your organisation's IT standard while allowing remote access for emergency / external support.
3. Make sure you server is configured to allow access to the other services in the DMZ(De Militarised Zone).
4. Note: Upto this point we are setting a standard Linux instance. Please feel free to make necessary security provision's as per your organisation's norms.
5. We assume you have already setup the following and know the credentials:
   a. SMTP server for mail
   b. MinIO for Storage
   c. MSSQL Server 2016 or above

(Note: These are products that are openly available and instructions to install and configure them are available with their respective vendors.)

6. Retrieve and have and have a copy of deployable binaries ready *(which is delivered digitally)*

1 Development Setup Instructions

Install Nodejs

https://nodejs.org

Install Angular CLI

Refer: https://angular.io/guide/setup-local#step-1-install-the-angular-cli

Run the command below in the root folder of the app via terminal to install dependencies

```
$ npm install
```

Dependencies-List

@angular/animations: ^7.2.15
@angular/cdk: ^7.3.7
@angular/common: ^7.2.15
@angular/compiler: ^7.2.15
@angular/core: ^7.2.15
@angular/forms: ^7.2.15

@angular/http: ^7.2.15
@angular/material: ^7.2.0
@angular/platform-browser: ^7.2.15
@angular/platform-browser-dynamic: ^7.2.15
@angular/router: ^7.2.15
@angular/upgrade: ^7.2.15
@fortawesome/angular-fontawesome: ^0.3.0
@fortawesome/fontawesome-svg-core: ^1.2.22
@fortawesome/free-brands-svg-icons: ^5.10.2
@fortawesome/free-solid-svg-icons: ^5.10.2
@ngx-share/button: ^7.1.4
@ngx-share/buttons: ^7.1.4
@ngx-share/core: ^7.1.4
@ngx-translate/core: ^11.0.1
@ngx-translate/http-loader: ^4.0.0
@progress/kendo-angular-pdf-export: ^1.3.1
@progress/kendo-drawing: ^1.5.12
@tinymce/tinymce-angular: ^3.3.0
@types/crypto-js: ^3.1.43
angular-in-memory-web-api : github:brandonroberts/in-memory-web-api-bazel#50a34d8
angular-progress-bar: ^1.0.9
angular-star-rating: ^4.0.0-beta.3
classlist.js: ^1.1.20150312
codeanalyzer: 0.0.1
codelyzer: ^5.1.0
core-js: ^2.5.4
crypto-js: ^3.1.9-1
css-star-rating: ^1.2.4
hammerjs: ^2.0.8
jszip: ^3.2.1
keycloak-angular: ^6.1.0
keycloak-signup-url: 0.0.2
mat-video: ^2.7.2
ng2-simple-timer: ^6.0.0
ngx-audio-player: ^7.1.5
ngx-bar-rating: ^1.1.0
ngx-bootstrap: ^5.2.0
ngx-chips: ^2.0.2
ngx-cookie-service: ^2.2.0
ngx-doc-viewer: ^0.1.20
ngx-image-viewer: ^1.0.13
ngx-owl-carousel: ^2.0.7
ngx-spinner: ^7.2.0

```
primeicons: ^1.0.0
primeng: ^7.1.3
quill: ^1.3.7
rxjs: ^6.3.0
rxjs-compat: ^6.5.3
tinymce: ^5.0.15
web-animations-js: ^2.3.2
zone.js: ~0.8.26
```

Update the environment file, path : src/environments/environment.ts

```
const keycloakConfig: KeycloakConfig = {
url: ’http://{keycloak_server_address}}/auth’, realm:
’{keycloak_realm_name}’, clientId: ’{keycloak_client_id}’, credentials:
{

secret: ’{keycloak_client_secret}’ }

};

// Add here your environment setup infos export const environment =
{ production: false, keycloak: keycloakConfig, clientUrl:
’http://localhost:4200’, adminUrl: ’http://{admin_app_URL}/’, apiUrl:
’http://{dotnet_api_address}/api/’, serverUrl:
’http://{dotnet_api_address}’, encKey: ’{random encryption key of
length 32(should be same as added in Angular Admin App
environment file settings)}’ };
```

Run the command below in the root folder of the app via terminal to run the app locally

```
$ ng serve --watch
```

2 Build for Deployment Instructions

- Install NodeJS
  - https://nodejs.org
- Install the Angular CLI
  - https://angular.io/guide/setup-local#step-1-install-the-angular-cli
- Run the command below in the root folder of the app via terminal to install dependencies

```
$ npm install
```

- Update the environment file, path : src/environments/environment.prod.ts

```
const keycloakConfig: KeycloakConfig = {
url: ’http://{keycloak_server_address}}/auth’, realm: ’{keycloak_realm_name}’,
clientId: ’{keycloak_client_id}’, credentials: {

secret: ’{keycloak_client_secret}’ }

};

// Add here your environment setup infos export const environment = {
production: false, keycloak: keycloakConfig, clientUrl: ’http://{client_app_URL}’,
adminUrl: ’http://{admin_app_URL}/’, apiUrl: ’http://{dotnet_api_address}/api/’,
serverUrl:

’http://{dotnet_api_address}’, encKey: ’{random encryption key of length
32(should be same as added in Angular Admin App environment file settings)}’ };
```

- Run the command below in the root folder of the app via terminal to build the app for Deployment

```
$ ng build --prod --build-optimizer --aot --output-hashing=none
```

- Distribution files will be generated in /dist folder

3 Deployment Instructions for Docker

Install Docker

First get and install Docker. Instructions are given on the Centos fourm based on your version or here https://www.docker.com/.

Create Dockerfile

Create a *text file* in a New folder in desktop, add the code below to the file and rename it as ’Dockerfile’. Remember to save it **without** any file extension.

- Create a text file in the root folder, add the code below to the file and rename it as ’Dockerfile’

```
FROM nginx:stable-alpine LABEL version="1.0"
COPY nginx.conf /etc/nginx/nginx.conf
WORKDIR /usr/share/nginx/html COPY ./ . EXPOSE 80
CMD ["nginx","-g","daemon off;"]
```

- Create a text file in the root folder, add the code below to the file and rename it as ’nginx.conf’

```
        worker_processes 1;
        events { worker_connections 1024; }
        http { server { listen 80; server_name localhost; root /usr/share/nginx/html;
        index index.html index.htm; include /etc/nginx/mime.types;
        gzip on; gzip_min_length 1000; gzip_proxied expired no- cache no-store private
        auth;
        gzip_types text/plain text/css application/json
        application/javascript application/x-javascript text/xml application/xml
application/xml+rss text/javascript;
        location / { try_files $uri $uri/ /index.html;
        } }
        }
```

- Run the command below in the root folder of the app via terminal to build docker image

  ```
  docker build -t {image name}
  ```

- Run the command via terminal to deploy the docker image

  ```
  docker run -d -p {deployment_port}:80 {image_name}
  ```