Lab 2 - Create new angular and add SSR – Hands On

**Step 1 – Create a new angular application**

ng new ssr-demo --style scss –routing

Open the project in your editor – vscode or your favorite ide

**Step 2 – Add basic pages and stuffs to it**

* Add routing
* Add given components
* Update app.component.html and styles
* Run ng serve to verify see results

Try to capture the load time of page using chrome performance extension or dev tools

<https://chrome.google.com/webstore/detail/page-performance/gembkfinllgmbkgbgdoaeopbahikjomp?utm_source=chrome-ntp-icon>

**Step 3 – Add SSR**

Run below command to add ssr

ng g universal –clientProject:ssr-demo

add following dependencies

* npm install -save @nguniversal/express-engine
* npm install -save @nguniversal/module-map-ngfactory-loader
* npm install -D ts-loader
* npm i webpack-cli

Update angular.json dist locations

**Step 4 – Create server.ts file**

Create a server.ts file and add configuration for express server to run application in server from dist directory

**Step 4 – Configuration**

* Copy given webpack.server.config.js
* Add following lines in package.json scripts
* "compile:server": "webpack --config webpack.server.config.js --progress --colors",
* "serve:ssr": "node dist/server",
* "build:ssr": "npm run build:client-and-server-bundles && npm run compile:server",
* "build:client-and-server-bundles": "ng build --prod && ng run ssr-demo:server:production --bundleDependencies all"

Add Following codes in main.server.ts

export { AppServerModule } from './app/app.server.module';

export { ngExpressEngine } from "@nguniversal/express-engine";

export { provideModuleMap } from "@nguniversal/module-map-ngfactory-loader";

export { renderModuleFactory } from '@angular/platform-server';

Add ModuleMap factory reference in app.server module

import { ModuleMapLoaderModule } from '@nguniversal/module-map-ngfactory-loader';

imports: [

AppModule,

ServerModule,

ModuleMapLoaderModule

],

**Step 5 – compile and run**

First build the dist bundles by running below command

npm run build:client-and-server-bundles

In Dist directory you can see two separate artifacts has been created for browser and server

then compile the server using below command

npm run compile:server

if everything is successfully executed then run

node dist/server

This will open up express server and will listen on port 4000. Open up browser and see the results.