2.2 Creating Libraries

This section will guide you to:

* Create custom Angular libraries
* Create a component in a library
* Use **ng new my-app export** library to the main application

This guide has mainly eight subsections, namely:

2.2.1 Installing Node JS

2.2.2 Installing Visual Studio Code

2.2.3 Installing Angular CLI

2.2.4 Creating a new Angular project

2.2.5 Creating a custom library

2.2.6 Creating a component in the library

2.2.7 Exporting the library and inline component to the main application

2.2.8 Pushing the code to GitHub repositories

**Step 2.2.1:** Installing Node JS

* Node.js is already installed in your lab. (Refer MEAN: Lab Guide - Phase 2)

**Step 2.2.2:** Installing Visual Studio Code

* Visual Studio Code is already installed in your lab. (Refer MEAN: Lab Guide - Phase 2)

**Step 2.2.3:** Installing Angular CLI

* Angular is already installed in your lab. (Refer MEAN: Lab Guide - Phase 2)

**Step 2.2.4:** Creating a new Angular Project

* Open Terminal
* Create a workspace by running the below command:

**mkdir my-workspace**

**cd my-workspace**

* Run the below command to create a new Angular project:

***ng new* my-app**

* Navigate to your app by running the below command:

**cd my-app**

* Run the application using the below command and then open **localhost:4200** in your browser:

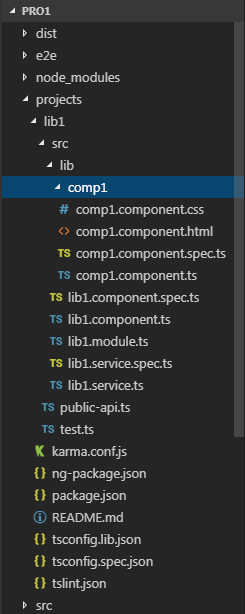
**ng serve**

**Step 2.2.5:** Creating a custom library

* **Please note:** Command to create the library would not work outside your project folder.
* Open the my-workspace folder in Visual Studio Code
* Open Terminal
* Go to my-app
* Run the below command inside your root folder:

**ng generate library my-lib**

* This will create a new folder with the name **projects** inside your main project's root folder. It will also contain a component with the same name as that of your library and necessary configuration files.

****

**Step 2.2.6:** Creating a component in the library

* Run the below command to create a component in your custom library:

**ng g component comp1**

**Step 2.2.7:** Exporting the library and inline component to the main application

* Go to my-app/projects/my-lib/src/lib
* Write the below code in **public-api.ts** file in your component folder:

/\*

\* Public API Surface of lib1

\*/

export \* from './lib/my-lib.service';

export \* from './lib/my-lib.component';

export \* from './lib/my-lib.module';

export \* from './lib/comp1/comp1.component';

* Write the below code in **<YourlibComponent>.module.ts** file:

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

import { Lib1Component } from './lib1.component';

import { Comp1Component } from './comp1/comp1.component';

@NgModule({

declarations: [Lib1Component, Comp1Component],

imports: [

],

exports: [Lib1Component, Comp1Component]

})

export class Lib1Module { }

* Write below code in **app.module.ts** file:

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

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

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

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

import { Lib1Module } from 'lib1';

@NgModule({

declarations: [

AppComponent

],

imports: [

BrowserModule,

AppRoutingModule,

Lib1Module

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

* Write the below code in **app.component.html** file:

<lib-lib1></lib-lib1>

<lib1-comp></lib1-comp>

* Run the project by executing the below command:

**ng serve**

**Step 2.2.8:** Pushing the code to GitHub repositories

Open your command prompt and navigate to the folder where you have created your files

**cd <folder path>**

Initialize your repository using the following command:

**git init**

Add all the files to your git repository using the following command:

**git add .**

Commit the changes using the following command:

**git commit . -m “Changes have been committed.”**

Push the files to the folder you initially created using the following command:

**git push -u origin master**