**Angular 4 - CLI**

**Installation**

Requirements

* Node Js (Latest Version, Doesn’t work on older version)
* NPM Latest Version

On Command Prompt

* Npm install –g @angular/cli (Install angular 4 cli)
* ng new Project\_Name (Creates new angular 4 project )
* cd Project\_Name (Go into project folder)
* ng serve --port 8089 (Starts server at 8089 port)
* ng –v (To check angular version)

Add UI Bootstrap

* npm install --save @ng-bootstrap/ng-bootstrap
* Add the lines in index.html

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-alpha.6/css/bootstrap.min.css" />

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

* Or install as node module

npm install bootstrap@4.0.0-alpha.6

* If installed as node module then angular-cli.json change “styles” to

"styles": [

"../node\_modules/bootstrap/dist/css/bootstrap.min.css",

"styles.css"

]

* In app.module.ts

import {NgbModule} from '@ng-bootstrap/ng-bootstrap';

@NgModule({

imports: […, NgbModule.forRoot()]

})

If you have src\systemjs.config.js

Add this in map

'@ng-bootstrap/ng-bootstrap': 'node\_modules/@ng-bootstrap/ng-bootstrap/bundles/ng-bootstrap.js',

**Important Informations**

For more info: [**https://ng-bootstrap.github.io/#/getting-started**](https://ng-bootstrap.github.io/#/getting-started)

For Bootstrap 4 : [**http://www.quackit.com/bootstrap/bootstrap\_4/tutorial/**](http://www.quackit.com/bootstrap/bootstrap_4/tutorial/)

**NB: Typescript only uses single quotes for string not double quotes**

**Data Binding**

Output Binding

* String Interpolation : {{data}}
* Property binding : [property] = “data” (any html property)

Input Binding

* Event Binding : (event) = “expression” (any html event without ‘on’ e.g. onClick becomes click, onInput becomes input)

Two-way binding

* [(ngModel)] = “data”

**Creating Component (WITHOUT CLI)**

* Create folder inside app folder E.g. navbar
* Create ts, html and css files navbar.component.ts, navbar.component.html, navbar.component.css
* Inside **navbar.component.ts**

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

Imports Component from angular/core package

* + Add Component tag

@Component({

selector: 'app-navbar',

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

styleUrls: ['./navbar.component.css']

})

* Selector should be unique and used to refer to the component
* templateUrl providespath to Url
  + Create class

export class NavbarComponent {

variableName ='Some Value';

}

**Registering Created Component In** ***app.module.ts***

* Import component

import { NavbarComponent } from './navbar/navbar.component';

* Declare Component

declarations: [

NavbarComponent,

AppComponent

]

**Creating Component (WITH CLI)**

ng generate component componentName **or** ng g c componentName

Creating Services (Without CLI)

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

@Injectable()

export class GetJsonService {

constructor() { }

anyFunction() {…….}

}

Creating Services (With CLI)

ng g s serviceName

NB: add anyFunction() {……} as above

Implementing Services in Component

* import

import { GetJsonService } from '../services/get-json.service';

* Dependency injection

@Component({

::::,

providers: [GetJsonService]

})

* Declare in constructor

constructor(private getJsonVar: GetJsonService) {

}

* Use service function

componentFunc() {

this. getJsonVar.anyFunction()

}

Adding HTTP calls to Services

* In app.module.ts

import { HttpModule } from '@angular/http';

imports: […, HttpModule ]

In Service file

import { Http, Response } from '@angular/http';

constructor(private http: Http) {}

**Communication with Components**

Input

Parent Side :

<app-childElement [text] = "value"></ app-childElement>

Child Side :

import { … , Input } from '@angular/core';

export class ClassName {

@Input() text: string;

:::

}

Output

Child Side :

import { … , Output, EventEmitter } from '@angular/core';

@Output() childFunc = new EventEmitter<string>();

Calling parent’s function from child’s function

someFunction(data) {

:::

this.childFunc.emit(data);

}

Parent Side :

<app-child (childFunc) = " parentFunc($event)" ></app- child>

parentFunc (data) {

:::

}

Between Components

**Routing**

Create components

Go to app.module.ts

Import RouterModule

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

Define Router

imports: [

…

…

RouterModule.forRoot([

{

path : 'dashboard', // Path Name

component : DashboardComponent // Component Name

}

])

]

**BUILD AND DEPLOY**

**Building (With angular CLI)**

ng build

Builds for dev

ng build --environment=prod --base-href .

builds for prod environment

Use –base-href . or go to dist folder after build and change base href tag to

<base href=".">

**Deploying**

**NOTE***: Npm install http-server before deploying*

1. Copy dist folder to anywhere
2. Open git bash on location
3. http-server ./dist