

Report Title

REPORT SUBTITLE

Name | Course Title | Date

# Angular Project Setup

This following command creates a new AngularProject with name AngularCrud:-

ng new AngularCrud --skip-tests true

Install Bootstrap by executing the following command

npm install bootstrap@3 --save

# Reading Data in Angular

export class Employee {

id: number;

name: string;

gender: string;

email?: string;

phoneNumber?: number;

}

Using Question mark, we represent the field as optional.

Use the following AngularCLI command to create ListEmployeesComponent. We will place all employee CRUD components in "employees" folder. This is the reason we prefixed the "employees" folder name in the command. Also, notice we have set --flat option to true as we do not want to place the ListEmployeesComponent files in it's own dedicated folder.

**ng g c employees/listEmployees --spec false --flat true**

The above command imported ListEmployeesComponent in app.module.ts and included it in the declarations @NgModule array.

# Angular Routing and Navigation

**Import the Router Module :**

Import the RouterModule into the application root module AppModule. The Router Module contains the Router service and Router directives such as (RouterLink, RouterLinkActive, RouterOutlet etc). So for us to be able to implement routing, we first need to import the Router Module in our AppModule. So in app.module.ts make the following changes. Notice the changes are commented and self-explanatory.

// Import RouterModule

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

// Include RouterModule in the "imports" array of the @NgModule() decorator

@NgModule({

  declarations: [...

  ],

  imports: [

    BrowserModule,

**RouterModule**

  ],

  providers: [],

  bootstrap: [AppComponent]

})

**Configure the application routes:**   
To configure routes, we first need to import Routes type from '@angular/router'. This Routes type is not required for the application to work. However, using it provides us intellisense and compile time checking. For example, mis-spelled properties of the Route object will be reported as errors.

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

// Each route maps a URL path to a component

// The 3rd route specifies the route to redirect to if the path

// is empty. In our case we are redirecting to /list

// pathMatch property value can be full or prefix. For now we

// will set it to full as we want to do a full match. In our upcoming videos,

// we will discuss the difference between prefix and full in detail.

const appRoutes: Routes = [

  { path: 'list', component: ListEmployeesComponent },

  { path: 'create', component: CreateEmployeeComponent },

  { path: '', redirectTo: '/list', pathMatch: 'full' }];

// To let the router know about the routes configured above,

// pass "appRoutes" constant to forRoot(appRoutes) method

// We also have forChild() method. We will discuss the difference

// and when to use one over the other in our upcoming videos

@NgModule({

declarations: [...

],

imports: [

  BrowserModule,

  RouterModule.forRoot(appRoutes)

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

# [Angular base href](https://www.youtube.com/watch?v=Std1QJpMEiE)

The browser uses the <base href> value to prefix relative URLs when referencing CSS files, scripts, and images.

During development we usually set this to a single forward slash as shown below.   
<base href="/">

During development, if you want a different base path other than "/", simply execute the "ng serve" command with --base-href option set to your desired base path as shown below.  
ng serve -o --base-href /emp/

On your local development machine, if you set the base href element in index.html to "/emp/" instead of a single "/" and if you run ng serve -o command without the "base-href" option  you will not see anything on the browser. When you open the browser developer tools, you will see the JavaScript bundle files failed to load. To fix this execute ng serve command along with the base href option as shown below.  
ng serve -o --base-href /emp/

On your local development machine, if you set the base href element in index.html to a single forward slash and if you want to deploy your application on a server on sub-folder called "emp", then you will have to remember to update the base href element value in index.html to "/emp/". There are 2 ways we can do this. 

1. Manually update the index.html file OR
2. Use the --base-href option along with the ng build command as shown below. This will update the "base href" element value index.html.ng build --base-href /emp/