Routing

While creating a project itself, we can add routing module to the project

ng new first-project –routing

when routing is added to the project, the project contains

node\_modules folder will contain routing libraries also

app-routing.module.ts file is added to the project

app.module.ts contains imports of AppRoutingModule

if you see app-routing.module.ts

const routes: Routes = [];

Steps:

1. app-routing.module.ts

const routes: Routes = [

  {path:"login",component:LoginComponent},

  {path:"product",component:ProductComponent}

];

1. create a menu component

ng g c menu

in menu.component.html,

<nav class="navbar navbar-expand-sm bg-dark navbar-dark">

<!-- Brand/logo -->

<a class="navbar-brand" href="#">Logo</a>

<!-- Links -->

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link" href="#">Link 1</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Link 2</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Link 3</a>

</li>

</ul>

</nav>

1. in app.component.html

in any of the columns, (where you want the routing outlet)

<router-outlet></router-outlet>

1. in menu component, change the hyper links to router links

<a class="nav-link" routerLink="login">Login</a>

HttpClient

What are the HTTP methods?

GET to retrieve info from the webservice

POST to add an entry to the webservice

PUT to update an existing entry in the webservice

DELETE to delete an existing entry in the webservice

PATCH to update one or more attributes of the entry in a webservice

GET all users

GET user by id

POST add an user

PUT modify an user

DELETE remove an user

<https://api.instantwebtools.net/v1/airlines>

The above url refers to a REST api that allows us to

GET, POST, PUT, DELETE

First, lets do these from a POSTMAN

We have to access above REST API from angular project.

So we import “HttpClientModule” in app.module.ts

Now, all the access to rest api, I want to write in a separate program.

Lets create a service class, that is going to access the rest api.

In this service class, we create our own functions/methods that can

Get

Post

Put

Delete

How to create a service?

ng g s airline

ng generate service airline

import { HttpClient } from '@angular/common/http';

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

@Injectable({

  providedIn: 'root'

})

export class AirlineService {

  URL='https://api.instantwebtools.net/v1/airlines';

  constructor(private http:HttpClient) { }

  retrieveAllAirlines()

  {

    return this.http.get(this.URL);

  }

  addAirline(airline:any)

  {

    return this.http.post(this.URL,airline);

  }

  findAirlineById(id:number)

  {

    return this.http.get(this.URL+"/"+id);

  }

}

While calling a http client we get response in asynchronous fashion.

So we need to subscribe these methods.

We have added new airline

Install Bootstrap and JQuery

install bootstrap ( go to current project location)

npm install --save bootstrap@4.0.0-beta.3

for latest stable version:

npm install --save bootstrap

now you can see that the node\_modules folder contains "bootstrap" folder

go to angular.json

angular.json file:

------------------

"styles": [

"src/styles.css",

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

npm install bootstrap jquery --save

angular.json file:

------------------

"styles": [

"src/styles.css",

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

],

"scripts": [

"node\_modules/jquery/dist/jquery.min.js",

"node\_modules/bootstrap/dist/js/bootstrap.min.js"

]

--------------------------------

If bootstrap.min.css won't work then

import this bootstrap.min.css in a style.css file like this:

/\* You can add global styles to this file, and also import other style files \*/

@import url(https://unpkg.com/bootstrap@4.1.0/dist/css/bootstrap.min.css)

@import url('https://unpkg.com/bootstrap@3.3.7/dist/css/bootstrap.min.css');

-----------------------------------------------------------------

<!doctype html>

<html>

<head>

<meta charset="utf-8">

<title>Angular Bootstrap Demo</title>

<base href="/">

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

<link rel="icon" type="image/x-icon" href="favicon.ico">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="anonymous">

</head>

<body>

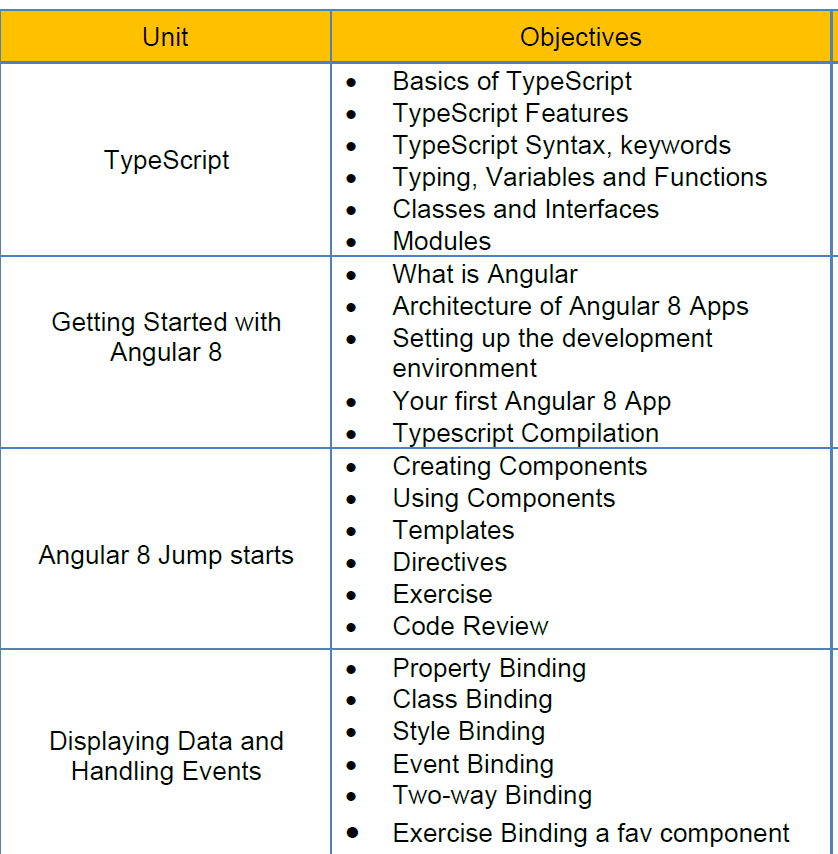
<app-root>Loading...</app-root>

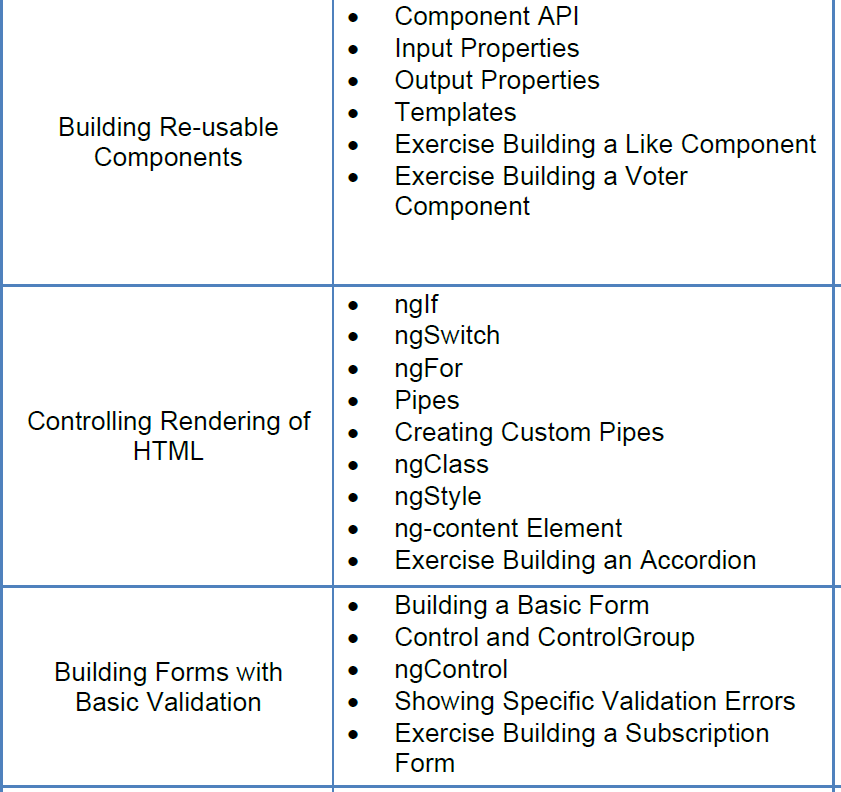
<script src="https://code.jquery.com/jquery-3.1.1.min.js" integrity="sha256-hVVnYaiADRTO2PzUGmuLJr8BLUSjGIZsDYGmIJLv2b8=" crossorigin="anonymous"></script>

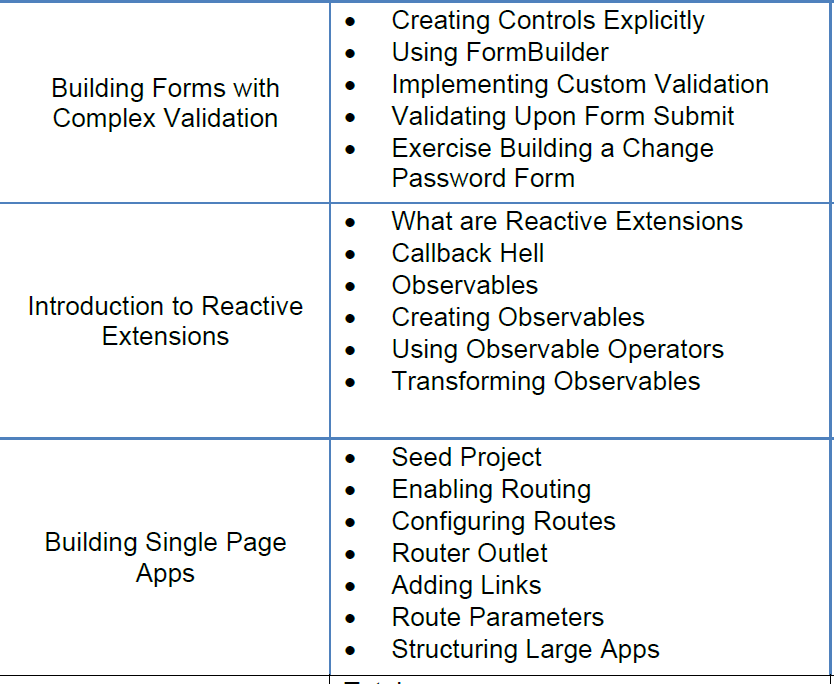
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js" integrity="sha384-Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA7l2mCWNIpG9mGCD8wGNIcPD7Txa" crossorigin="anonymous"></script>

</body>

</html>







Reactive forms support validations.

<div class="row">

    <div class="col-sm-3"></div>

    <div class="col-sm-3">

        <form [formGroup]="airlineForm">

            Id: <input type="number" name="" id="" [ngClass]="{'form-control':true,'is-invalid':c.id.touched && !c.id.valid}" formControlName="id" >

            Name: <input type="text" name="" id="" [ngClass]="{'form-control':true,'is-invalid':airlineForm.controls.name.touched && !airlineForm.controls['name'].valid}" formControlName="name"><span \*ngIf="airlineForm.controls.name.touched && !airlineForm.controls.name.valid" class="alert-danger">                Name is mandatory</span>

<br>            Country: <input type="text" name="" id="" class="form-control" formControlName="country">

            Logo: <input type="text" name="" id="" class="form-control" formControlName="logo">

            Slogan: <input type="text" name="" id="" class="form-control" formControlName="slogan">

            Head Quaters: <input type="text" name="" id="" class="form-control" formControlName="head\_quaters">

            Website: <input type="text" name="" id="" class="form-control" formControlName="website">

            Established: <input type="number" name="" id="" class="form-control" formControlName="established">

            <br>

            <input type="button" value="Add" class="btn btn-success" (click)="fnAdd()">

        </form>

         <br>

        {{airlineForm.controls.id.valid}}

    </div>

    <div class="col-sm-3"></div>

</div>

<div class="row">

    <div class="col-sm-3">

        <table class="table table-bordered table-striped table-hover">

            <thead>

                <tr>

                    <th>Id</th>

                    <th>Name</th>

                    <th>Country</th>

                    <th>Logo</th>

                    <th>Slogan</th>

                    <th>Head Quaters</th>

                    <th>Website</th>

                    <th>Established</th>

                </tr>

            </thead>

            <tbody>

                <tr \*ngFor="let a of airlines">

                    <td>{{a.id}}</td>

                    <td>{{a.name}}</td>

                    <td>{{a.country}}</td>

                    <td>

                        <img [src]="a.logo" alt="" width="100">

                    </td>

                    <td>{{a.slogan}}</td>

                    <td>{{a.head\_quaters}}</td>

                    <td>{{a.website}}</td>

                    <td>{{a.established}}</td>

                </tr>

            </tbody>

        </table>

    </div>

</div>

  constructor(private as:AirlineService, private fb:FormBuilder) {

    this.airlineForm=this.fb.group({

      id:['',[Validators.required]],

      name:['',Validators.required],

      country:[],