ng new recipe-book –prefix rb

Just delete the spec file.

We need the bootstrap file

**Creating the header :**

Open cli and move to recipe-book\src\app

Create the header by : ng g c header –flat (Not create new folder but create the component in the same folder)

For creating the bootstrap navbar goto : <http://getbootstrap.com/components/#navbar.Copy> and paste the item.

<app-header></app-header>

<div class=”container”>

<div class=”row”>

<div class=”col-lg-12”>

<app-recipes></app-recipes>

<app-shopping-list></app-shopping-list>

</div>

</div>

</div>

**recipes.component.html**

<div class="row">

<div class="col-md-5">

List is here

</div>

<div class="col-md-7">

Recipe Detail

</div>

</div>

**shopping-list.component.html**

<div class="row">

<div class="col-lg-10">

<app-shopping-edit></app-shopping-edit>

<hr>

The List

</div>

</div>

**header.html**

<nav class="navbar navbar-default">

<div class="container-fluid">

<div class="navbar-header">

<a href="" class="navbar-brand">Recipe Book</a>

</div>

<div class="collapse navbar-collapse">

<ul class="nav navbar-nav">

<li><a href="">Recipes</a></li>

<li><a href="">Shopping List</a></li>

</ul>

<ul class="nav navbar-nav navbar-right">

<li class="dropdown">

<a href="#" class="dropdown-toggle" role="button">Manage</a>

<ul class="dropdown-menu">

<li>Save Data</li>

<li>Fetch Data</li>

</ul>

</li>

</ul>

</div>

</div>

</nav>

**Go to recipes component class :**

recipes = []

**recipe.model.ts**

export class Recipe

{

public name : string ;

public description : string;

public imagePath : string;

constructor(name : string,desc : string,imagePath : string){

this.name = name;

this.description = desc;

this.imagePath = imagePath;

}

}

**recipe-list.html**

<div class="row">

<div class="col-lg-12">

<button class="btn btn-success">Add Recipe</button>

</div>

</div>

<hr>

<div class="row">

<div class="col-lg-12">

<a href="" class="list-group-item clearfix" \*ngFor="let recipe of recipes">

<div class="pull-left">

<h4 class="list-group-item-heading">{{recipe.name}}</h4>

<p class="list-group-item-text">

{{recipe.description}}

</p>

</div>

<span class="pull-right">

<img src="{{recipe.imagePath}}" alt="" class="img-responsive" style="max-height:50px;"/>

</span>

</a>

<app-recipe-item></app-recipe-item>

</div>

</div>

**recipe-detail.html**

<div class="row">

<div class="col-lg-12">

<img src="" class="img-responsive">

</div>

</div>

<div class="row">

<div class="col-lg-12">

<h1>Recipe Name</h1>

</div>

</div>

<div class="row">

<div class="col-lg-12">

<div class="btn-group">

<button class="btn btn-primary dropdown-toggle">Manage Recipe</button>

<ul class="dropdown-menu">

<li>To shopping list</li>

<li>Edit Recipe</li>

<li>Delete Recipe</li>

</ul>

</div>

</div>

</div>

<div class="row">

<div class="col-lg-12">

Description

</div>

</div>

<div class="row">

<div class="col-lg-12">

Ingredients

</div>

</div>

**Communication between the components :**

**Shopping List :**

ingredient.model.ts inside app/shared/ingredient.model.ts

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

import { Ingredient } from '../shared/ingredient.model';

@Component({

selector: 'app-shopping-list',

templateUrl: './shopping-list.component.html',

styleUrls: ['./shopping-list.component.css']

})

export class ShoppingListComponent implements OnInit {

ingredients : Ingredient[]=[

new Ingredient('Apples',5),

new Ingredient('Bananas',12)

];

constructor() { }

ngOnInit() {

}

}

Edit component :

<div class="row">

<div class="col-lg-12">

<form>

<div class="row">

<div class="col-lg-5">

Name

<input class="form-control">

</div>

<div class="col-lg-2">

Amount

<input class="form-control">

</div>

</div> <br>

<div class="row">

<div class="col-lg-12">

<button class="btn btn-success">Add</button>

<button class="btn btn-delete">Delete</button>

<button class="btn btn-primary">Clear</button>

</div>

</div>

</form>

</div>

</div>

**Routing in recipe :**

Create app-routing.module.ts in root directory

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

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

import {RecipesComponent} from './recipes/recipes.component';

import {ShoppingListComponent} from './shopping-list/shopping-list.component';

const appRoutes : Routes = [

{path:'',redirectTo:'/recipes',pathMatch : 'full'},

{path : 'recipes',component : RecipesComponent},

{path : 'shopping-list',component : ShoppingListComponent}

];

@NgModule({

imports : [RouterModule.forRoot(appRoutes)],

exports : [RouterModule]

})

export class AppRoutingModule

{}

Make it’s entry inside the app.module.ts

imports: [

BrowserModule,

FormsModule,

HttpModule,

**AppRoutingModule**

]

**Apply the link on header.component.html**

<li routerLinkActive=”active”><a routerLink="/recipes">Recipes</a></li>

<li routerLinkActive=”active”><a routerLink="/shopping-list">Shopping List</a></li>

Note : Router link active is to make the link active.

**Now , for the child routing**

const appRoutes : Routes = [

{path:'',redirectTo:'/recipes',pathMatch : 'full'},

{path : 'recipes',component : RecipesComponent,children : [

{path : '',component : RecipeStartComponent},

{path : 'new',component : RecipeEditComponent},

{path : ':id',component : RecipeDetailComponent},

{path : ':id/edit',component : RecipeEditComponent}

]},

{path : 'shopping-list',component : ShoppingListComponent}

];

@NgModule({

imports : [RouterModule.forRoot(appRoutes)],

exports : [RouterModule]

})

**Inside the recipe-edit.component.ts**

constructor(private route : ActivatedRoute,private recipeservice : Recipeservice) { }

ngOnInit() {

this.route.params.subscribe(

(params : Params)=>{

this.id = +params['id'];

this.editMode=params['id'] !=null;

console.log("id is "+this.editMode);

}

)

}

**When clicked on edit Recipe from the recipe detail.**

<button class="btn btn-primary" (click)="editRecipe()">Edit Recipe</button>

Inside the Component we do :

constructor(private recipeservice : Recipeservice,private router : Router,private route : ActivatedRoute) { }

editRecipe(){

this.router.navigate(['../',this.id,'edit'],{relativeTo : this.route});

}

**When clicked on the add recipe :**

<button class="btn btn-success" (click)="addRecipe()">Add Recipe</button>

Inside the Component we Have :

constructor(private recipeservice : Recipeservice,private router : Router,private route : ActivatedRoute) { }

addRecipe(){

this.router.navigate(['new'],{relativeTo : this.route});

}

**Observable**

It is only a data source :

Just an object that import from 3rd party.

Here we have the observable and the observer.

Events , Http requests

**Observer**

Handle error , Handle data , Handle completion (Hooks)

Observable are different approach of handling that (ie promises,callback functions)

The main advantages of the observables is there operators .

Ex :

ngOnInit() {

this.route.params.subscribe((params : Params)=>{

this.id = +params['id'];

this.recipe=this.recipeservice.getRecipe(this.id);

},

()=>{},

()=>{}

)

Here data is sent by the angular and handled by the observable.

Inside the subscribe method we have the observer part.

**Building first observable :**

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

import {Observable} from 'rxjs/Observable';

import 'rxjs/Rx';

ngOnInit() {

const myNumbers = Observable.interval(1000);

myNumbers.subscribe(

(number:number)=>{

console.log(number)

}

);

}