Angular

**Javascript array functions:**

**Map:**

const array1 = [1, 4, 9, 16];

// pass a function to map

const map1 = array1.map(x => x \* 2);

console.log(map1);

// expected output: Array [2, 8, 18, 32]

**Find:**

const array1 = [5, 12, 8, 130, 44];

const found = array1.find(element => element > 10);

console.log(found);

// expected output: 1

**Filter:**

const words = ['spray', 'limit', 'elite', 'exuberant', 'destruction', 'present'];

const result = words.filter(word => word.length > 6);

console.log(result);

// expected output: Array ["exuberant", "destruction", "present"]

**Creating an Angular project:**

Cd folder

ng new angularcourse

running Angular project:

cd angularcourse

ng serve –o

**creating a new component:**

ng generate component new\_component1

to insert new\_component to app.component:

**go to** app.component and **insert** <app- new\_component1></app- new\_component1> .

**creating a class:**

create a folder named ‘shared’ under src/app

create a file named **dish.ts**:

export class Dish{

id:String;

name:String;

image:String;

price:String;

}

Import the class in new\_component1.components.ts:

import {Dish} from '../shared/dish';

**Creating a class containing another class:**

Create a file dish.ts:

**import { Comment } from './/comment';**

export **class Dish** {

id: string;

name: string;

image: string;

category: string;

featured: boolean;

label: string;

price: string;

description: string;

comments: **Comment[];**

}

Before we should create a Comment class:

export **class Comment** {

rating: number;

comment: string;

author: string;

date: string;

}

**Creating an object to be exported:**

Dishes.ts:

**export const DISHES**: Dish[] = [

{

id: '0',

…

**creating an object + ngFor:**

in **new\_component1.components.ts**:

export class MenuComponent implements OnInit {

dishes: Dish[]=[

{

id:'0', // this is a dish instance

name:'Pizza',

image:'/assets/images/uthappizza.png',

price:'1.9'

},

{

id:'1',

name:'Vadonut',

image:'/assets/images/vandonut.png',

price:'2.6'

}

];

In new\_component1.components.html:

**<div \*ngFor="let i of dishes">**

<img matListAvatar src={{i.image}} alt={{i.name}}>

<p>{{i.name}}</p>

**</div>**

**Or:**

<ul>

**<li \*ngFor="let i of dishes">**

<img src={{i.image}}>

<p>{{i.name}}</p>

**</li>**</ul>

**Or:**

**<p \*ngFor="let i of dishes">**

{{i.name}}

**</p>**

**ngIf:**

in new\_component.ts:

selected\_dish=dishes[0];

in new.componen.html:

**<div \*ngIf="selectedDish"> // it’s hidden when selesctedDish variable is empty**

<p>you have choosen the best dish</p>

**</div>**

**Hidden:**

<div **[hidden]="dishes"**> **// will be hidden if dhishes variable is not empty**

<mat-spinner></mat-spinner><h4>Loading . . . Please Wait</h4> </div>

**Data binding and Event binding:**

Dishdetail.component.ts:

export class component{

@Input toto: string;

…

}

<app-dishdetail **[toto]="this.selectedDish"** ></app-dishdetail>

selectedDish is an object existing in **dishdetail.component.ts.**

**MenuComponent.ts:**

export class MenuComponent implements OnInit {

dishes=DISHES;

selectedDish:Dish;

constructor() { }

ngOnInit(): void { }

**onSelect(x:Dish){ // x is an object in the array Dish**

**this.selectedDish=x;**

**}**}

**MenuComponent.html:**

<div fxFlex>

<mat-grid-list cols="2" rowHeight="200px">

<mat-grid-tile **\*ngFor="let x of dishes" (click)="onSelect(x)"** >

<img height="200px" src="{{x.image}}">

<mat-grid-tile-footer>

<h1>{{x.name | uppercase}}</h1>

</mat-grid-tile-footer>

</mat-grid-tile>

</mat-grid-list>

</div>

**Explanation:**

On click on the image of mat-grid-tile the attribute selectedDish become equal to x. So, we export it to dishdetail component.

To accept this object we should import the “import” module in **dishdetail.component.ts**:

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

export class DishdetailComponent implements OnInit {

@Input() **Object-to-input**:Dish; Dish is the type of dish variable

…

**Services**

**Creating a Service:**

**Ng generate service** **folder\_name/service\_name**

After creating the service **it should be imported into app.module.ts**:

import{ServiceName} from './/services/servicename.service';

**providers: [serviceNameService]**

**message.service.ts:**

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

@Injectable({

providedIn: 'root'

})

export class MessageService {

constructor() { }

**hello\_message():string{**

**return "hello in my web site";**

**}**

}

**importing a service into a component:**

**In the componentname.component.ts:**

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

**import{MessageService} from '../services/message.service';**

export class MenuComponent implements OnInit {

**msg:string; // declaring the attribute**

constructor(**private toto:MessageService**) {

}

ngOnInit(): void {

**this.msg=this.toto.hello\_message();**

}

// msg="hello in my web site"

**Promise:**

const one = new Promise<string>((resolve, reject) => {

resolve('Hello');

});

one.then(value => {

console.log('resolved', value);

});

one.catch(error => {

console.log('rejected', error);

});

**Rx/JS: a service that returns an observable:**

**Implementing and getting the observable:**

**import { Observable,of } from 'rxjs';**

import{delay} from 'rxjs/operators';

…

Implementing the observable:

**getDishes():Observable<Dish[]>{**

**return of(DISHES);**

**}**

Getting the observable:

this.dishService.getDishes().subscribe(response => this.dishes = response);

**SwitchMap function:**

switchMap give an observable from an other observable.

import { switchMap } from 'rxjs/operators';

**Subscribe:**

we use subscribe after a function that return an observable.

ErrorMessage:string;

this.Service.getMessage().subscribe((response)=>this.file=Message,

(error)=>this.ErrorMessage='the server is not sending response');

**Routing:**

**Create a file named app.routing.ts:**

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

**import { HomeComponent }** from './/home/home.component';

**import { AboutComponent }** from './/about/about.component';

**import{MenuComponent**} from './/menu/menu.component';

**const app\_routing: Routes = [**

**{ path: 'home',** **component: HomeComponent },**

**{ path: 'menu', component: MenuComponent },**

**{ path: ' ', redirectTo: '/home', pathMatch: 'full' } // it is obligatory to configure a redirectTo with pathMatch:’full’ the default page otherwise the router will not work**

**];**

**export const ROUTING=RouterModule.forRoot(app\_routing);**

**import this constant to app.module.js:**

import{ROUTING} from './/app.routing';

imports: [

BrowserModule,

ROUTING,

…

**Update app.component.html:**

<app-header></app-header>

**<router-outlet></router-outlet>**

<app-footer></app-footer>

**To go to “menu”:**

<a mat-button routerLink="/menu">menu</a>

Design:

<a mat-button routerLink="/menu" routerLinkActive="toto"

>menu</a>

.toto {backgroundcolor:red} // to color the active link with red

**Another way to route:**

Add the following code to ROUTING constant after importing DishdetailComponent in **app.routing.ts**:

**{ path: 'dishdetail/:link', component: DishdetailComponent },**

We have already dishdetail component in menu component, we add **[routerLink]="['/dishdetail',x.id]"** in **menu.component.html:**

<div \*ngFor="let **x** of dishes **" [routerLink]="['/dishdetail',x.id]"** >

<img height="200px" src="{{x.image}}">

<h1>{{x.name | uppercase}}</h1>

</div>

**<app-dishdetail ></app-dishdetail>**

**In dishdetail.component.ts:**

import {Router,ActivatedRoute} from '@angular/router';

export class DishdetailComponent implements OnInit {

**dish:Dish;**

constructor(private ar: ActivatedRoute) { }

ngOnInit(): void {

**this.ar.params.subscribe**(

(response) => {

this.dish = response.**link**;

});

}

**Dishdetail.component.html:**

<div fxFlex \*ngIf="dish">

<h3 \*ngIf="dish">{{dish.name | uppercase}}</h3>

<p>{{dish.description}}</p

</div>

**Result:**

Localhost:4200/dishdetail/0: contain the dish 0

Localhost:4200/Dishdetail/1:…….

**To open a component in a pop-up:**

We would from the header component open the login component:

In app.module.ts:

import { MatDialogModule } from '@angular/material/dialog';

and:

imports:[

…

**MatDialogModule**

],

**entryComponents: [**

**LoginComponent**

],

**In header.component.ts :**

import { MatDialog, MatDialogRef } from '@angular/material/dialog';

…

**openLoginForm() {**

**this.dialog.open(LoginComponent, {width: '500px', height: '450px'}); // appeal the login component**

**}**

**In header.component.html :**

We put a button which appel openLoginFormFunction:

<a mat-button (click)="openLoginForm()"> Login</a>

**In login.component.html:**

<mat-toolbar color="primary">

Login

<span class="flex-spacer"></span>

<button mat-button mat-dialog-close>&times;</button> // &times is the “X” exit icon

</mat-toolbar>

**To save a form in a json object:**

**HTML:**

<p>{{ user | json }}</p>

<form novalidate **(ngSubmit)="onSubmit()"**> // non validate: It’s not HTML 5 who will validate the formular but angular will hundle the validation

**<mat-dialog-content>**

<p>

<mat-form-field>

<input matInput type="text" **[(ngModel)]="user.username" name="username"**>

</mat-form-field>

<mat-form-field>

<input matInput placeholder="Password" type="password" **[(ngModel)]="user.password" name="password"**>

</mat-form-field>

<mat-checkbox **[(ngModel)]="user.remember" name="remember"**>Remember Me</mat-checkbox>

</p>

**</mat-dialog-content>**

**<mat-dialog-actions>**

<button mat-button mat-dialog-close>Cancel</button>

<**button type="submit"**>Login</button>

**</mat-dialog-actions>**

</form>

**TS :**

**import {MatDialog, MatDialogRef} from '@angular/material/dialog';**

export class LoginComponent implements OnInit {

user = {username: '', password: '', remember: false};

constructor(public dialogRef: **MatDialogRef<LoginComponent>**) { } // dialogRef function closes the loginComponent

ngOnInit() { }

onSubmit() {

**this.dialogRef.close();**

}}

**Reactive Forms :**

Importing the Reactive Forms Module:

import { MatSelectModule } from '@angular/material/select';

import { MatSlideToggleModule } from '@angular/material/slide-toggle';

import { ReactiveFormsModule } from '@angular/forms';

…

**Creating the feedback class:**

export class Feedback {

firstname: string;

lastname: string;

telnum: number;

email: string;

agree: boolean;

contacttype: string;

message: string;

};

export const ContactType = ['None', 'Tel', 'Email'];

TS:

import **{ FormBuilder, FormGroup**, Validators } from '@angular/forms';

import { Feedback, ContactType } from '../shared/feedback';

…

export class ContactComponent implements OnInit {

**feedbackForm: FormGroup; // declaring the feedback variable**

feedback: Feedback;

contactType = ContactType;

constructor(**private fb: FormBuilder**) {

this.createForm(); // create an empty form

}

ngOnInit() {

}

createForm() {

this.feedbackForm = this.fb.group({ // create the feedback form

firstname: '',

lastname: '',

telnum: 0,

email: '',

agree: false,

contacttype: 'None',

message: ''

});

}

onSubmit() {

***this.feedback = this.feedbackForm.value; // feedback will receive the value of the inputs***

this.feedbackForm.reset();

}

}

**HTML:**

<form novalidate **[formGroup]="feedbackForm"** **(ngSubmit)="onSubmit()**">

<mat-form-field class="half-width">

<input matInput **formControlName="firstname"** placeholder="First Name" type="text">

</mat-form-field>

<mat-form-field class="half-width">

<input matInput **formControlName="lastname"** placeholder="Last Name" type="text">

</mat-form-field>

<button type="submit" mat-button class="background-primary text-floral-white">Submit</button>

</form>

**HTTP/REST:**

**Installing json server to emulate a real server:**

Cd project\_folder

npm install json-server -g

json-server --watch db.json

or json-server --watch db.json -d 2000 (with a delay of 2 seconds)

**db.json:**

[{

"nom":"blouza",

"prix":"10D",

"path":"assets/folder/images/img-pro-01.jpg",

"link":"/a-propos-de-nous"

},

{ "nom":"blouza",

"prix":"10D",

"path":"assets/folder/images/img-pro-01.jpg",

"link":"/a-propos-de-nous"}]

**GET method:**

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

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

@Component({

selector: 'app-aboutus',

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

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

})

export class AboutusComponent implements OnInit {

id:string;

baseURL:string;

result:string;

constructor(private ar:ActivatedRoute,private http: HttpClient) { }

ngOnInit(): void {

this.baseURL="http://localhost:3000/db";

this.http.get(this.baseURL)

.subscribe(x => {

this.response = x;

this.result=this.response["enfant"][0].nom; display blouza

});

}

}

**POST:**

interface Article {

id: number;

title: string;

}

this.http.post<Article>('https://jsonplaceholder.typicode.com/posts', { title: 'Angular POST Request Example' }).subscribe(data => {

this.postId = data.id;

})

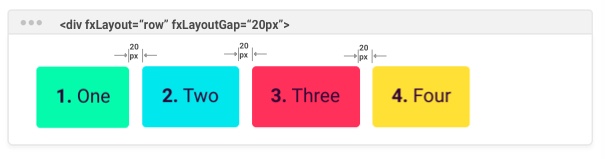
**Design with Angulal/Material:**

**Div fxLayout:**

<div fxLayout="row" fxLayoutGap="20px">

<div>1. One</div> <div>2. Two</div> <div>3. Three</div> <div>4. Four</div>

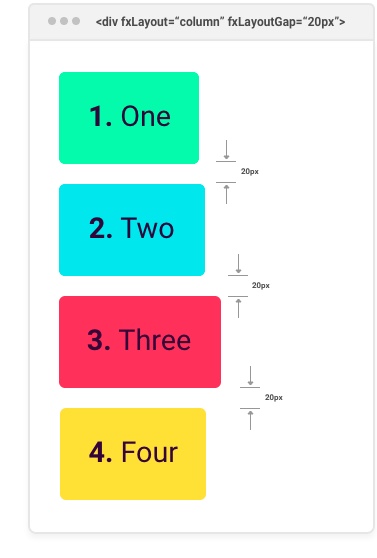
</div>



<div fxLayout="column" fxLayoutGap="20px">

<div>1. One</div> <div>2. Two</div> <div>3. Three</div> <div>4. Four</div>

</div>



**MatgridList:**

**<mat-grid-list cols = "4" rowHeight = "100px">**

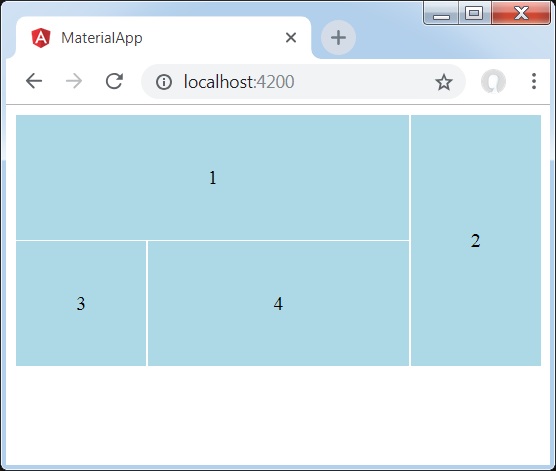
**<mat-grid-tile** [colspan] = "3" [rowspan] = "1">1 **</mat-grid-tile>**

<mat-grid-tile [colspan] = "1"[rowspan] = "2">2</mat-grid-tile>

<mat-grid-tile [colspan] = "1" [rowspan] = "1">3</mat-grid-tile>

<mat-grid-tile [colspan] = "2" [rowspan] = "1">4 </mat-grid-tile>

**</mat-grid-list>**



**Mat-Card:**

**<mat-card>**

**<mat-card-header>**

**<mat-card-title>**

<h3>{{selectedDish.name | uppercase}}</h3>

**</mat-card-title>**

**</mat-card-header>**

<img mat-card-image src={{selectedDish.image}} alt={{selectedDish.name}}>

**<mat-card-content>**

<p>{{selectedDish.description}}

</p>

**</mat-card-content>**

**<mat-card-actions>**

<button mat-button>LIKE</button>

<button mat-button>SHARE</button>

**</mat-card-actions>**

**</mat-card>**



npm install --save @angular/material @angular/cdk

ng add @angular/material