

Day 3 Exercises

Exercise 1

Time: 15 mins.

1. Install Angular
 - I. use the npm install command from your Command Line Interface: **npm install -g @angular/cli**
2. Verify your installation
 - I. on the command line: **ng version**
3. Create a project named store using Angular CLI
 - I. on the command line: **ng new store**
4. Review the files created for your project and start the ng server
 - I. On the command line change to the “store” directory
 - II. start the server using npm start

You should see the following:

```
charly@worker01:~/Angular/store$ npm start
> store@0.0.0 start
> ng serve

✓ Browser application bundle generation complete.

Initial Chunk Files | Names      | Raw Size
vendor.js           | vendor     | 2.02 MB |
polyfills.js        | polyfills  | 315.26 kB |
styles.css, styles.js | styles    | 207.81 kB |
main.js             | main       | 50.10 kB |
runtime.js          | runtime    | 6.51 kB |
                    | Initial Total | 2.59 MB

Build at: 2022-07-07T16:09:43.490Z - Hash: c0cb09a4fb4883ed - Time: 37055ms

** Angular Live Development Server is listening on localhost:4200, open your browser on
http://localhost:4200/ **

✓ Compiled successfully.
✓ Browser application bundle generation complete.

Initial Chunk Files | Names      | Raw Size
main.js             | main       | 7.23 kB |
runtime.js          | runtime    | 6.51 kB |
```

5. Review the application on your browser and your phone

6. Clear the app.component.js and create a label and a input text element that modifies the label when the text changes.

- I. On app.module.js

```
import { FormsModule } from '@angular/forms';
imports: [
    BrowserModule,
    AppRoutingModule,
    FormsModule
],
```
- II. On app.component.js

```
{{title}}
<input type="text" [(ngModel)] = "title"/>
```

Exercise 2

Time: 15mins.

Create your fake data

1. Install json-server
 - I. use the command `npm install -g jsonserver`
2. create a `server` folder at the top level of the project
3. download the `db.json` file and put it on the `server` folder
4. modify the package.json to add a script named "serverAPI"

```
I. {
  "name": "store",
  "version": "0.0.0",
  "scripts": {
    "ng": "ng",
    "start": "ng serve",
    "build": "ng build",
    "watch": "ng build --watch --configuration development",
    "test": "ng test",
    "serverAPI": "json-server --watch server/db.json --port 3000"
  },
  "private": true,
  "dependencies": {
    "@angular/animations": "^14.0.0",
    "@angular/common": "^14.0.0",
    "@angular/compiler": "^14.0.0",
    "@angular/core": "^14.0.0",
    "@angular/forms": "^14.0.0",
```

```

"@angular/platform-browser": "^14.0.0",
"@angular/platform-browser-dynamic": "^14.0.0",
"@angular/router": "^14.0.0",
"rxjs": "~7.5.0",
"tslib": "^2.3.0",
"zone.js": "~0.11.4"
},
"devDependencies": {
"@angular-devkit/build-angular": "^14.0.4",
"@angular/cli": "~14.0.4",
"@angular/compiler-cli": "^14.0.0",
"@types/jasmine": "~4.0.0",
"jasmine-core": "~4.1.0",
"karma": "~6.3.0",
"karma-chrome-launcher": "~3.1.0",
"karma-coverage": "~2.2.0",
"karma-jasmine": "~5.0.0",
"karma-jasmine-html-reporter": "~1.7.0",
"typescript": "~4.7.2"
}
}

```

Exercise 3

Time 5 minutes.

Install Angular Material.

Modify the header component to use the toolbar component of Angular Material.

Answer

From the Command Line Interface of your OS (in the root directory of your project):

```
store $ ng add @angular/material
```

Respond yes to the typography question and select your preferred theme.

header.component.ts

```

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

@Component({
  selector: 'app-header',
  template: `
<mat-toolbar color="primary">
<span>My Store</span>
</mat-toolbar>
`
})

```

```
styleUrls: ['./header.component.scss']
})
export class HeaderComponent {}
```

app.component.html

```
<app-header></app-header>
```

Exercise 4

Time 15 mins.

Create a routing to a component that will talk about your store, call it “about” and configure the app to display it behind the header when

Answer:

About.component.ts

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

@Component({
  selector: 'app-about',
  templateUrl: './About.component.html',
  styleUrls: []
})

export class AboutComponent implements OnInit {
  constructor() {}
  ngOnInit(): void {
  }
}
```

About.component.html

```
<p>Hello World!</p>
```

app-routing.module.ts

```
import { AboutComponent } from './pages/About.component';
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';

const routes: Routes = [
  {path: 'about',
  component: AboutComponent}
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
```

```
exports: [RouterModule]
})
export class AppRoutingModule { }
```

app.component.html

```
<app-header></app-header>
<router-outlet></router-outlet>
```

Exercise 5

Time 1 hour

Create a products module, a product component, a product interface and a product services and make them all work together.

1. Create a products module with the route to get it displayed on the application when we call the “products” directory inside of our URL
2. Create a product component inside the products module, it should display when the URL “products/product” is called.
3. Given the structure of the db.json file create an interface for the product type of objects
4. Given the URL of the json server we create on [Exercise 2](#) create a service to get the products in the json-server
5. On the products component import the service and include the code to write the products on the console

Answer

```
ng g m pages/products -m=app --route products
ng g c pages/products/product -m=products --route product
ng g s pages/products/services/products
```

product.service.ts

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';
import { Product } from '../interfaces/product.interface';

@Injectable({
  providedIn: 'root'
})
export class ProductsService {
  private jsonServerURL="http://localhost:3000/products";
  constructor(private httpClient: HttpClient) { }

  getProducts(): Observable<Product[]> {
```

```

return this.httpClient.get<Product[]>(this.jsonServerURL);
}
}

```

product.interface.ts

```

import { AppRoutingModuleModule } from "src/app/app-routing.module";

export interface Product{
  id: number;
  name: string;
  price: number;
  description: string;
  categoryId: number;
  stock: number;
}

```

products.component.ts

```

import { Component, OnInit } from '@angular/core';
import { ProductsService } from '../services/products.service';
import { tap } from 'rxjs';

@Component({
  selector: 'app-products',
  templateUrl: './products.component.html',
  styleUrls: ['./products.component.scss']
})
export class ProductsComponent implements OnInit {

  constructor(private queryService: ProductsService) { }

  ngOnInit(): void {
    this.queryService.getProducts().pipe(
      tap(res => console.log(res))
    ).subscribe();
  }
}

```

Exercise 6

Time 30 mins

Create the product page displaying the products with Angular Material components and put a button to add the product in the shopping cart

Answer

Product.components.scss

```
section.products{
display: flex;
flex-wrap: wrap;
justify-content: center;
align-items: center;
margin-top: 30px;
.card{
margin: 15px;
min-height: 100px;
min-width: 300px;
max-width: 300px;
height: 100%;
}
.button{
width: 100%;
}
.mat-card-header{
justify-content: center;
text-justify: center;
font-weight: bold;
}
}
```

Material.module.ts

```
import { NgModule } from "@angular/core";
import { MatToolbarModule } from "@angular/material/toolbar";
import { MatCardModule } from "@angular/material/card";
import { MatButtonModule } from "@angular/material/button";
import { MatIconModule } from "@angular/material/icon";

@NgModule({
  exports: [MatToolbarModule, MatCardModule, MatButtonModule, MatIconModule]
})

export class MaterialModule{}
```

products.component.html

```
<section class="products">
<mat-card class="card" *ngFor="let product of productList">
<mat-card-header>
{{product.name}}
</mat-card-header>
<mat-card-content>
{{product.description}}<br>
```

```

{{product.price | currency}}
</mat-card-content>
<mat-card-actions>
<button mat-flat-button color="primary">Add to the cart
<mat-icon>shopping-basket</mat-icon>
</button>
</mat-card-actions>
</mat-card>
</section>

```

products.component.ts

```

import { Component, OnInit } from '@angular/core';
import { ProductsService } from './services/products.service';
import { tap } from 'rxjs';
import { Product } from './interfaces/product.interface';

@Component({
  selector: 'app-products',
  templateUrl: './products.component.html',
  styleUrls: ['./products.component.scss']
})
export class ProductsComponent implements OnInit {
  productList!: Product[];

  constructor(private queryService: ProductsService) { }

  ngOnInit(): void {
    this.queryService.getProducts().pipe(
      tap((actualProducts: Product[]) => this.productList=actualProducts)
    ).subscribe();
  }
}

```

Exercise 7

Time 30 mins

Create an icon on the toolbar to show the number of products and the total amount of the products on the shopping cart.

Answer

Create a folder named services under the shared folder and create a cart service, then use the service

Create a new component to show the cart data.

Add the service result to the new component

Incorporate the new component to the header

cart.service.ts

```
import { Injectable } from "@angular/core";
import { TitleStrategy } from "@angular/router";
import { Observable, Subject } from "rxjs";
import { Product } from "src/app/pages/products/interfaces/product.interface";
@Injectable({providedIn:'root'})
export class CartService{
  products : Product[] = [];
  private cartSubject = new Subject<Product[]>();
  private totalSubject = new Subject<number>();
  private quantitySubject = new Subject<number>();
  get cartAction$() : Observable<Product[]>{
    return this.cartSubject.asObservable();
  }
  get totalAction$() : Observable<number>{
    return this.totalSubject.asObservable();
  }
  get quantityAction$() : Observable<number>{
    return this.quantitySubject.asObservable();
  }
  private cartTotal():void{
    const total=this.products.reduce((acc, product)=> acc+=product.price, 0);
    this.totalSubject.next(total);
  }
  private productsQuantity():void{
    const total = this.products.length;
    this.quantitySubject.next(total);
  }
  private addToCart(product:Product):void{
    this.products.push(product);
    this.cartSubject.next(this.products);
  }
  newProductToTheCart(product:Product):void{
    this.addToCart(product);
    this.cartTotal();
    this.productsQuantity();
  }
}
```

cart.component.ts

```
import { NgIf } from "@angular/common";
import { Component } from "@angular/core";
import { CartService } from "src/app/shared/services/cart.service";
```

```

@Component({
  selector:'app-cart',
  template:`
    <ng-container *ngIf = '{total : total$ | async , quantity : quantity$ |async} as dataCart'>
    <ng-container *ngIf = '{products : products$ |async}'>
    <mat-icon>shoppig_cart</mat-icon>
    {{total$ | async | currency}}
    {{quantity$ | async | json}}
    </ng-container>
    </ng-container>
  `
})
export class CartComponent{
  quantity$ = this.cartService.quantityAction$;
  total$ = this.cartService.totalAction$;
  products$ = this.cartService.cartAction$;
  constructor(private cartService : CartService){}
}

```

header.component.scss

```

.spacer{
  flex: 1 1 auto;
}

```

product.services.ts

```

import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';
import { Product } from '../interfaces/product.interface';

@Injectable({
  providedIn: 'root'
})
export class ProductsService {
  private jsonServerURL="http://localhost:3000/products";
  constructor(private httpClient: HttpClient) { }

  getProducts(): Observable<Product[]> {
    return this.httpClient.get<Product[]>(this.jsonServerURL);
  }
}

```

product.component.ts

```

import { Component, OnInit } from '@angular/core';
import { ProductsService } from '../services/products.service';

```

```
import { tap } from 'rxjs';
import { Product } from './interfaces/product.interface';
import { CartService } from 'src/app/shared/services/cart.service';

@Component({
  selector: 'app-products',
  templateUrl: './products.component.html',
  styleUrls: ['./products.component.scss']
})
export class ProductsComponent implements OnInit {
  productList!: Product[];

  constructor(private queryService: ProductsService, private cartService: CartService) { }

  ngOnInit(): void {
    this.queryService.getProducts().pipe(
      tap((actualProducts: Product[]) => this.productList=actualProducts)
    ).subscribe();
  }

  addToCart(product: Product):void{
    this.cartService.newProductToTheCart(product);
  }
}
```