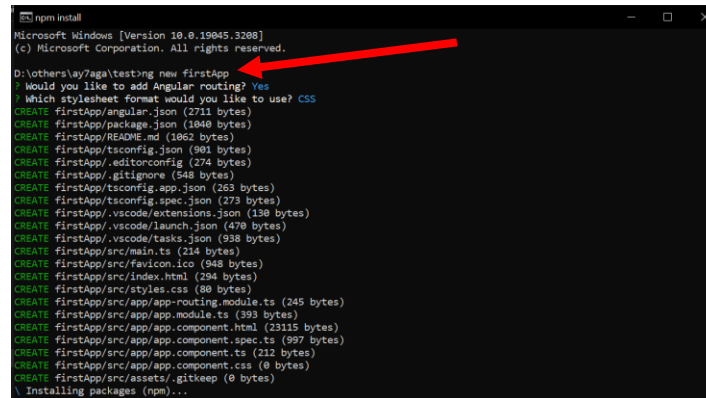


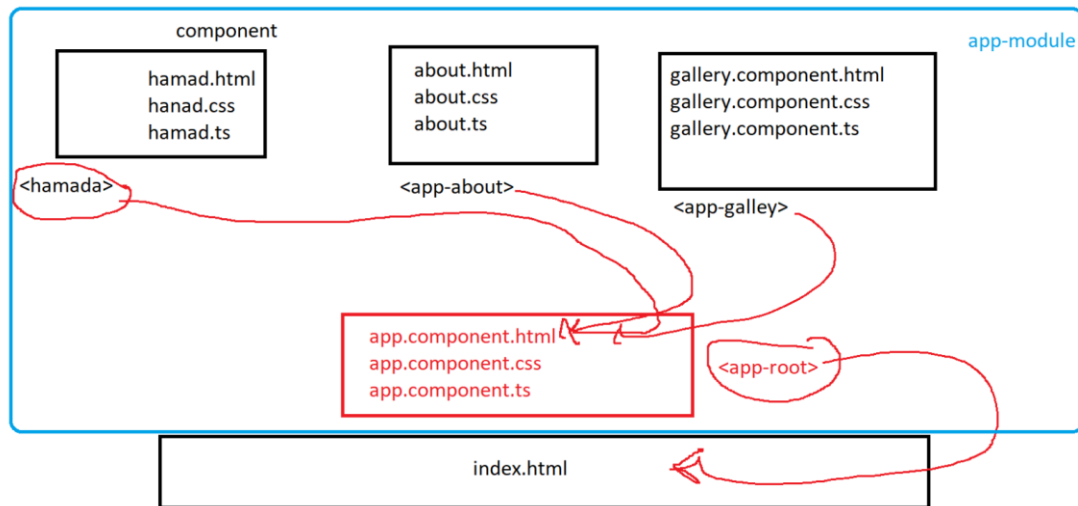
- الـ Angular هي كمبونت بيز استراكتشر عبارة عن مجموعة من الكمبونتس والكمبونتس دي اللي بترن البروجكت بتاعى.
- عشان اشغل Angular اول مرة على الجهاز عندى.
 - تحميل Node js من جوجل وعمل install لها Next>Next.. من خلال اللينك دا <https://nodejs.org/en/download>
 - افتح cmd واكتب فيها الـ command line دا `npm i -g @angular/cli` الخطوة دي بعملها مرة واحدة في حياتي.
 - بعد كدا هحتاج اعمل project بعمل folder يشيل البروجكت دا وادخل جوا الفولدر واسليكت على path الفولدر من فوق واكتب cmd وادوس انتر واكتب الـ command line دا `ng new firstApp`



```
Microsoft Windows [Version 10.0.19045.3208]
(c) Microsoft Corporation. All rights reserved.

D:\others\ay7aga\testing new firstApp
C:\> ng new firstApp
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? CSS
CREATE firstApp/angular.json (2711 bytes)
CREATE firstApp/package.json (1040 bytes)
CREATE firstApp/README.md (1062 bytes)
CREATE firstApp/tsconfig.json (981 bytes)
CREATE firstApp/.editorconfig (274 bytes)
CREATE firstApp/.gitignore (540 bytes)
CREATE firstApp/tsconfig.app.json (263 bytes)
CREATE firstApp/tsconfig.spec.json (273 bytes)
CREATE firstApp/.vscode/extensions.json (130 bytes)
CREATE firstApp/.vscode/launch.json (470 bytes)
CREATE firstApp/.vscode/tasks.json (938 bytes)
CREATE firstApp/src/main.ts (214 bytes)
CREATE firstApp/src/favicon.ico (948 bytes)
CREATE firstApp/src/index.html (294 bytes)
CREATE firstApp/src/styles.css (80 bytes)
CREATE firstApp/src/app/app-routing.module.ts (245 bytes)
CREATE firstApp/src/app/app.module.ts (393 bytes)
CREATE firstApp/src/app/app.component.html (23115 bytes)
CREATE firstApp/src/app/app.component.spec.ts (997 bytes)
CREATE firstApp/src/app/app.component.ts (212 bytes)
CREATE firstApp/src/app/app.component.css (0 bytes)
CREATE firstApp/src/assets/.gitkeep (0 bytes)
\ Installing packages (npm)...
```

- بعد كدا عشان اشغل البروجكت هكتب `cd firstApp` وادوس انتر وبعد كدا اكتب `code` وادوس انتر عشان افتح البروجكت ككود انما لو انا عايز افتح البروجكت على الـ browser هكتب `ng serve --open` وادوس انتر او اكتب `ng s --o`.
- بعد ما يفتح البروجكت فى vsCode هحتاج افتح الـ Terminal بتاعته ادوس على '+' ctrl هتفتح معايا وممكن افتح البروجكت على البروزر من هنا وهنا هعمل فيها حاجات كتير هنشوفها بعد كدا. لو في ايورر ظهر ليك هتحله بالفيديو دا <https://www.youtube.com/watch?v=kS3sBTa-bKw>
- دي صورة بتوضحلي الاستراكتشر بتاع الـ Angular. الـ Angular عبارة عن components.



- نفتح البروجكت هنلاقي بداخله فولدرات وحاجات كثيرة في منها اللي هحتاجه وفي منها اللي مش هحتاجه تعال نشوفهم.
- فولدر ال src وهو اهم فولدر ودا من الاخر دا البروجكت بتاعى .. هفتحه هلاقي حاجات كتير وبسيطة وهنفهمهم مع بعض
 - o فولدر ال app هو الفولدر اللي بيحتوي علي ال components بتاعتي وال Services واللوجيك وال Routing وكل حاجة.
- ملحوظة لو عايز اوقف ال Server اكتب C^ في ال terminal . ولو عايز السطر السابق في ال commend ادوس سهم لفوق .

components

- ال Angular ينقسم الكود الى Components يعنى مواضيع وال Component الواحد لازم يحتوي علي 3 ملفات اساسيين .html,css,ts.
- name.component.html - name.component.css - name.component.ts
- كل Component بيكون له ال Selector بتاعه.
- ال Angular عامله Component اسمه app.component وال Component دا الوحيد اللي له صلاحية التعامل مع ال index وهو بيكون وسيط بين ال Components بتاعتي و ال index لان ال index لازم يكلم Component واحد بس ودا الصح عشان ميتخانقوش مع بعض والسليكتور بتاع ال app.component اسمه <app-root>

How to Create Component from cli

- نكتب في ال terminal بتاعتي ال VSCode نكتب ng g c << ng g c nameComponent aboutComponent
- تعريف الاختصارات دي generate يعني اعمل انشاء
- ng = Angular , g = generate , c = component

- 2- هنادي السليكتور بتاعى بقا فى app.component.html والسليكتور هيكون ب اسم الكموننت اللى انا كريتته (انشاءته) وهنكتبه كدا syntax
<app-nameComponent></app-nameComponent> مثال <app-about></app-about>
- ng g c HomeComponent –SKIP tests لو كتبت ال Command دا معناه اني مش عايز ملف ال tester اللى بيعملهولي مع الكموننت

شرح الموجود فى ملف ال **TypeScript** اللى بيتم إنشائه تلقائي مع ال **Component**

```
import {Component} from '@angular/core';
@Component({
  selector: 'app-home-component',
  templateUrl: './home-component.component.html',
  styleUrls: ['./home-component.component.css']
})
export class HomeComponentComponent {
}
```

- طبعا مش فاهم حاجة (🤔) ف تعال بقا نشرح كل سطر فيهم

```
import {Component} from '@angular/core'; // هنا بقوله اعمل امبورت لكلاس الكموننت
الموجود فى فولدر الكور الموجود فى الانجولر
//Decorator bta3t component or metadata bta3t component الداتا التعريفية بتاعت
الكموننت
@Component({ // object
  selector: 'app-home-component', // بروبرتي اسمها سليكتور اقدر من خلالها اسليكت او استخدم
الكموننت بتاعى فى اي مكان
  templateUrl: './home-component.component.html', // بروبرتي بتشاور على الباث
بتاع الاتش تي ام ال بتاع الكموننت بتاعى
  // بروبرتي بتشاور على الباث بتاع ملف ال سي اس اس بتاعى وممكن اشاور على اكثر من ملف سي اس اس
styleUrls: ['./home-component.component.css']
})
//والكموننت بيكون عبارة عن كلاس بعمله اكسبورت
export class HomeComponentComponent {
  // هنا بكتب الكود بتاعى بقا
}
```

- ملحوظة : الكلاس اللى بييجي بعد الميئا داتا بتاعت الكموننت هو اللى هيمثل الكموننت لان هو الدايركت كلاس اللى بعد الميئا داتا ولذلك يجب مراعاة الترتيب عند اضافة كلاس اخر فى نفس الملف.
- ملحوظة : ليه بنعمل run لل Angular علي Server ؟ الاجابة عشان اقدر استخدم ال Import , Export اللى بنستخدمهم فى ال .ts

شرح الموجود في ملف الـ `app.module.ts`

`app.module.ts` هو عبارة عن container يحتوي على مجموعة من الـ Component. والكود التالي دا اللي بيكون موجود في

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';

@NgModule({
  declarations: [
    AppComponent,
    HomeComponentComponent
  ],
  imports: [
    BrowserModule,
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

- طبعا مش فاهم حاجة (🤔) فد تعال بقا نشرح كل سطر فيهم

```
import { NgModule } from '@angular/core'; // كل الامبورتنس دول مش محتاج اشرحهم
import { BrowserModule } from '@angular/platform-browser';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';

// Decorator bta3 Al App-Module
@NgModule({
  declarations: [ // بعرف في اي كمبوننت جديد انا عملته
    AppComponent,
  ],
  imports: [ // لو انا بستخدم موديول ثاني مع الالاب موديول بضيف اسم الموديول هنا
    BrowserModule,
    AppRoutingModule
  ],
  providers: [ // لو انا عامل سيرفس باجي اضيفها هنا
    // ليس له علاقة بالبوتستراب العادي وهو مرتبط القصة بتاعتنا ولو انا عندي 20 كمبوننت باجي اكتب هنا اسم
    // الكمبوننت اللي انا هعمله سايكتور في الاندكس
  ],
  bootstrap: [AppComponent]
```

```
})  
export class AppModule { }
```

- ملحوظة : ليه في الـ index بضيف selector واحد ؟ الاجابة عشان اقدر اعمل فكرة الـ Routing.

install bootstrap

- هكتب في الـ terminal كذا `npm I bootstrap` ثم بعد ذلك هفتح ملف اسمه `angular.json` عشان اربط البوتستراب جلوبال.
- هضيفهم في اول array اسمه `scripts` , `style` لان اللي تحت خالص دول خاصين بالـ `testing`. وهضيفهم بالشكل دا

```
"styles": [  
  "./node_modules/bootstrap/dist/css/bootstrap.min.css",  
  "src/styles.css"  
],  
"scripts": [  
  "./node_modules/bootstrap/dist/js/bootstrap.bundle.min.js"  
]
```

- وهكذا في font awesome و الـ jQuery.
- ملحوظة مهمة جدا عند عمل اي تعديل على `angular.json` لازم اوقف السيرفر واشغله تاني عشان التعديل يسمع معاها.
- `angular.json` هو ملف بيحتوي علي كل الـ Configuration اللي هتكون جلوبال على كل البروجكت.

install fontawsome

```
npm i @fontawesome/fontawesome-free
```

ثم بعد ذلك نفتح ملف `angular.json` ونضيف في ملف الـ سي اس اس بتاعها

```
"./node_modules/@fontawesome/fontawesome-free/css/all.min.css",
```

install jQuery

```
npm install jquery
```

ثم بعد ذلك نفتح ملف `angular.json` ونضيف في ملف الـ JavaScript بتاع الـ jQuery

```
"./node_modules/jquery/dist/jquery.min.js",
```

install angular Material >> <https://material.angular.io/>

هكتب في التيرمنال كذا `-- ng add @angular/material`

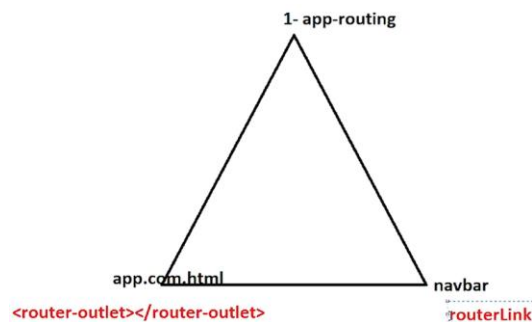
هيسألني اسالة هوافق عليها كلها عادي

```
-- in app.module.ts >>
import {MatSlideToggleModule} from '@angular/material/slide-toggle';
imports: [
  MatSlideToggleModule,
]
-- in your component you need add this >>
<mat-slide-toggle>Toggle me!</mat-slide-toggle>
```

وبعد كذا هنشغل السيرفر تانى لانه بيحتاج اننا نشغله من تانى ng serve
وبعد كذا نفتح ال site بتاع angular material ونروح للتايه اللي اسمها components واختار بقا اللي انا عايزه واخذ الكود بتاعه وال import بتاعه

Routing

اهميته هي التنقل بين الصفحات بدون عمل load للصفحة. وله 3 حاجات اساسية وهى



app-routing -1

app.component.html -2

<router-outlet></router-outlet>

navbar -3

RouterLink = 'pathName'

خطوات عمل Routing :

1. open >> app-routing.module.ts file then

in routes Array Add your routing as object

like this >> `{path: 'home', component: HomeComponentComponent}`

2. يوجد اتربيوت فى ال Angular اسمه RouterLink وبيتم اضافته كـ Attribute فى ال navbar الموجودة فى ال navbar مثلاً ودي طريقة كتابته.

RouterLink = 'pathName' like RouterLink = 'home'

```
<li class="nav-item my-1">
  <a class="nav-link" RouterLink="home">home<a>
</li>
```

وبكدا لما اكلك علي اللينك دا هيفتحلي الكمبوننت بتاعي

-- وعندنا اتريبوت تاني اسمه

RouterLinkActive = 'addClass' like 'bg-danger' or custom class.

- وعندنا property اسمها redirectTo بتقوله لو ال path فاضي اعمل routing على اسم ال path دا وساعتها لما يشوف اسم ال path هيقوم رايح لل component بتاعه على طول ويشغله.
- وعندنا property تانيه اسمها pathMatch وهو بديله شرط يظهر على اساسه يعني لو قولتله pathMatch 'full': هنا بقوله لازم الكلمة اللي مكتوبة في ال redirectTo تساوي الكلمة المكتوبة في ال path الموجود مع ال component اما لو قولتله 'prefix': pathMatch هنا مش لازم تكون الكلمتين شبه بعض لو اول الكلمة شبهه خلاص هتشتغل .. والعادة اننا بنستخدم full وتعال نوضح بمثال

```
{path:'', redirectTo:'home', pathMatch:'full'},
{path:'home', component : HomeComponentComponent},
```

- وعندي حاجة جميلة برضو وهي ان لو ال user كتب حاجة في ال path فوق مش موجودة عندي ومش عامل ليها routing هيروح علي طول على ال routing اللي انا ضايفه له .. وملحوظة الترتيب بيفرق معاها ولازم تكون اخر حاجة تحت خالص في ال routing مثال.

```
{path:'**', component:NotFoundComponentComponent}
```

- عندنا Property اسمها title بتقوله لما اكون واقف في ال component دا خليلي اسم الصفحة من فوق خالص في التابه كدا يكون بالاسم اللي انا كاتبه في ال title .

Session 02

Child Routing

يمكنك من خلاله عمل routing داخل ال routing الاساسي مثال لو انا عندي routing اسمه product ومحتاج اضيف بداخله routing للموبايل مثلا و routing لللاب توب مثلا بحيث اني لما ادخل على ال product الاقي الموبايل واللاب توب وادوس على ال mobile يشغلي ال routing بتاعه مثال

```
{path:'Product', component : productComponentComponent,
  children:[
    {path:'mobile', component : mobileComponentComponent},
    {path:'laptop', component : laptopComponentComponent}
  ]
},
```

ومتتناسش تروح تضيف ال routerLink بقا في ال html بتاع ال product

- معلومات سريعة احنا عرفينها كدا : الكلاس هو عبارة عن مجموعة من ال properties ومجموعة من ال method ال property : هي اني محطش قبلها var or let واكتب اسم ال property على طول.

- Assets folder الملف دا بيكون جوا البروجكت بتاعك ودا اعتبره المطبخ بتاعك تحط في اي صور انت هتستخدمها او نعمل ملف جافا اسكريبت واربطه او فيديو وهكذا.

- لو انا عايز اضيف صورة في الانجلر بكتب كدا عشان اوصل للصور الموجودة في ال assets

- مينفعش اعمل for loop علي object . مش هينفع خالص تمام واصلا مفيش حاجة اسمها object في ال angular تمام

Binding

معناها بالعربي ربط او دس وهى اني بربط داتا من ال ts جوا ال html code وله ككذا نوع وهما

1- Data Binding << وهى انى بعمل داتا بايند يعني بدس داتا في ال html

<h1>welcome {{username}} from home</h1>

2- Property Binding << وهى اني بعمل بروبرتي بايند يعني بدس داتا في اتربيوت

3- Event Binding << وهو اني ببايند فانكشن جوا الايفنت بتاعى

<button (click)="test()"></button>

واله 3 انواع دول كلهم بنسميهم one way data binding يعنى بأخد داتا من ال ts بوديها الى html بس كدا

Data Binding -1

امثلة بكود كامل :

```
// in class component
export class ViewClientInvoicesComponentComponent {
  userName:string = "Ali Mostafa";
} // in html code
h1>hello {{userName}}</h1>
```

Property Binding -2

```
// in class component
imgPath:string = "../../assets/images/p2.jpg";
<!-- in HTML Code -->
<img [src]="imgPath">
```

event Binding -3

```
// in class component
test(){
  alert("Hello Ahmed");
} <!-- in HTML Code -->
<button (click)="test()">click Here</button>
```


directives ngFor

بكل بساطة لو عندى array في ال ts وعايز اخده عندى فى ال html بستخدم ngFor او لو انا عايز اعمل loop علي حاجة معينة بستخدمها برضو. مثال

```
// in class component
ourEmp:string[] = ["Ahmed","Ali","Hager"];
<!-- in HTML Code -->
<h3 *ngFor="let x of ourEmp"> {{x}} </h3>
```

ملحوسة : بضيف ال *ngFor في ال Tag اللى انا عايزه يتكرر يعنى لو انا عايز اقرر صفوف في جدول هضيفها فى ال <tr> بتاعت الجدول مثال سريع

```
<tr *ngFor="let client of ourClients">{{client}}</tr>
```

object

```
// in class component
ourClients:object = {eName:"BMW",Id:1,Modal:2023};
<!-- in HTML Code -->
<h3>{{ourClients.Name}}</h3> // Error
```

لو انا قولتله كذا هيجبلي Error لانه هو بيبيص علي ال object الاساسي بتاع ال type Script عشان احنا قولنا ان ال Name موجوده عنده ولا لا وهنا هو مش هيلاقبها ويجبك ال error دا

Property 'Name' does not exist on type 'object' والحل بتاعها هو اني اعمل **interface** وهنا يجي الدور بتاع ال **Interface** بقا يلا بينا نعمله

Interface

هعمله مؤقتا فى ال ts بتاعت ال home ولازم اضيفه قبل ال @component لان الكلاس لازم يكون لازم فيهما ع طول

```
interface Client{
  Name:string;
  Id:number;
  modal:number;
}
@Component({.....})

// in class component
ourClients:Client = {Name:"BMW",Id:1,Modal:2023};
<!-- in HTML Code -->
<h3>{{ourClients.Name}}</h3>
```

طيب انا ممكن بكل بساطة بدل ما بضيف ال interface في ال home.ts ومش هقدر استخدمه فى اي component ثاني اقدر اعمله فى ملف لوحده منفصل واعملها Export عشان اقدر استخدمه فى اي حته ثانيه والحته الثانيه اضيف فيها import وننشأه من خلال Terminal نكتب ng g i client + انتر

```
// in file interface
export interface client {
  Name:string;
  Id:number;
  Modal:number;
}

// in class component
ourClients:client = {Name:"BMW",Id:1,Modal:2023};
// بمجرد ما بكتب كلاينت هو بيعمل امبورت تلقائي فوق وبيكون شكله كذا
import { Emp } from '../emp';
<!-- in HTML Code -->
<h3>{{ourClients.Name}}</h3>
```

يلا نطبق بمثال حلو على بعض الحاجات اللي فاتت

```
// in interface
export interface ViewClientInvoicesInterface {
  codeInvoices:number;
  Paid:number;
  RemainingAmountOf:number;
  InvoiceDetails:string;
}

// in viewClient.ts
export class ViewClientInvoicesComponentComponent {
  viewInvoicesData:ViewClientInvoicesInterface[] =
  [
    {codeInvoices:234545,Paid:1000 ,RemainingAmountOf:
500,InvoiceDetails:"ahmed"},
    {codeInvoices:334521,Paid:1000 ,RemainingAmountOf:
500,InvoiceDetails:"ahmed"},
    {codeInvoices:444558,Paid:1000 ,RemainingAmountOf:
500,InvoiceDetails:"ahmed"},
    {codeInvoices:554545,Paid:1000 ,RemainingAmountOf:
500,InvoiceDetails:"ahmed"},
    {codeInvoices:374535,Paid:1000 ,RemainingAmountOf:
500,InvoiceDetails:"ahmed"}
  ]
}

<!-- in HTML Code -->
<table class="table">
  <thead>
    <tr>
      <th scope="col" class="mainBGolor">تفاصيل الفاتورة</th>
```

```

<th scope="col" class="mainBGolor">المتبقي</th>
<th scope="col" class="mainBGolor">المدفوع</th>
<th scope="col" class="mainBGolor">كود الفاتورة</th>
</tr>
</thead>
<tbody>
<tr *ngFor="let invoice of viewInvoicesData">
<td>{{invoice.InvoiceDetails}}</td>
<td>{{invoice.RemainingAmountOf}}</td>
<td>{{invoice.Paid}}</td>
<td>{{invoice.codeInvoices}}</td>
</tr>
</tbody>
</table>

```

Directives index

يعنى لو انا عايز اظهر ال index بتاع ال array بتاعى فى كل مرة ال for بتلف فيها

```

<tr *ngFor="let invoice of viewInvoicesData; let code = index">
<td>{{code +1}}</td> // +1 to start from 1 2 3 4 5 6

```

directives even odd

ومن خلالها ببسأل هل الصف دا زوجي ولا فردي ف لو انا قولتله = even كذا هيحطلي true قدام الصف اللى رقمه زوجي و false قدام الصف اللى رقمه فردي والعكس كذلك لو قولتله odd هيحط true قدام الرقم الفردي و يحط false قدام الرقم الزوجي .

الاستفادة منها ممكن اعمل حاجة بناء على كونديشن معين مثلا اقله لو الصف دا true اعمل كذا غيرلي الباكر جروند مثلا

```

<tr *ngFor="let invoice of viewInvoicesData ; let x = even"> or x = odd
<td>{{x}}</td>

```

Even	متبقي الفاتورة	المدفوع	المتبقي	كود الفاتورة
true	⊕	500	1000	234545
false	⊖	500	1000	334521
true	⊕	500	1000	444538
false	⊖	500	1000	554545
true	⊕	500	1000	374535

Directives if conditions

بكل بساطة بتحكم فى ال div او اي شئ بناء على شرط معين يعني مثلا فى المثال القادم هتحكم فى اظهار واخفاء ال div بناء على شرط وهو لو كان true هيطهره ولو قولتله false هيجفي.

```

<div *ngIf="true">Ahmed</div> // كذا هيطهرلي كلمة احمد انما لو قولتله فوولس مش هيطهر كلمة
احمد تمام

```

طيب تعال نطبق دا على مثال حيوي اكر

لو انا عايز اتحكم فى ظهور واخفاء الجدول مثلا بناء على ان الجدول في داتا ولا لا فلو الجدول به داتا اظهرهولي ولو مفيش اظهرلي مكانه حاجة تانيه.

```

<table *ngIf="viewInvoicesData.length > 0 ; else MessageNoDataTable " >
  ... <!-- لو الشرط اتحقق هيطهر الجدول انما لو متحققش هيطهرلي التاج اللي تحت دا -->
</table>
// to show message if table no data ... this tag is implemented in
Angular
<ng-template #MessageNoDataTable>
  <div class="text-center text-danger fw-bold pt-5 mt-5">
    
  </div>
</ng-template>

```

two way data binding هي لو انا عايز اعمل عكس ال Binding اللي فاتت وهي لو انا عندي داتا في ال html وعايز اوديها في ال ts

ودا بيتم عن طريق حاجة اسمها [(ngModel)] واسمها المشهور هو BANANA IN A BOX فهو بالظبط ضرب عصفورين بحجر في انه عمل شكل الايفنت و وفي نفس الوقت خلاه يتعامل مع الفاريبول الموجود في ال ts وبتشتغل مع ال inputs وبتقوله انت بقا يا انبوت هتأثر في انهي فاريبول وقتها بقا بقوله = اسم الفاريبول بتاعى وفي المثال القادم اكنى هعمل key up ف اي حاجة هكتبها في الانبوت هتظهرلي في المكان اللي انا بعمل في بايندينج للفاريبول وملحوظة لازم اعمل Import لل module بتاعها في ال app-routing واسمه **FormsModule**. تطبيق المثال

```

// in ts create variable:
userName : string = "" ;

<!-- in HTML -->
<input [(ngModel)]="userName" class="form-control mx-auto my-5 w-50">
<div>welcome : {{userName}}</div>

// Decorator bta3 Al App-Module
imports: [ // لو انا بستخدم موديول ثاني مع الالاب موديول بضيف اسم الموديول هنا
  BrowserModule,
  AppRoutingModule , FormsModule // add FormsModule
],

```

Class Binding style binding Directive ngClass Directive ngStyle

لو انا عايز اضيف class او Style بناء على Condition معين . تعريف اخر <> لو انا بلعب في استايل تاجات معينة بناء علي variables.

Class Binding

```
// يوجد طريقتين يمكنك استخدامهم
```

```

    // in ts
    pPrice:number = 9000;
    <div [class.bg-danger] = "pPrice<6000"> // اول طريقة
        Ahmed Mostafa
    </div>
    // ثاني طريقة اكني بستخدم اف والس ويكتب بطريقة تيرناري اوبريتور
    // in ts
    pPrice:number = 9000;
    <div [class]="pPrice<6000? 'bg-danger': 'bg-white'"> // ternary operator
        Ahmed Mostafa
    </div>

```

style Binding

```

// يوجد طريقتين يمكنك استخدامهم
// in ts
pPrice:number = 9000;
<div [style.color]="pPrice < 6000? 'red':'green' " >
    Ahmed Mostafa
</div>
// ثان طريقة اكني بستخدم اف والس ويكتب بطريقة تيرناري اوبريتور
// in ts
pPrice:number = 9000;
<div [style]="pPrice > 6000 ? 'color:red; background-color:white': 'color:green' ">
    Ahmed Mostafa
</div>

```

Directive ngClass

لو انا عايز اغير فى الاستايل بناء على اكثر من شرط

```

<div [ngClass]="{
    'bg-success': pPrice >5000 && pPrice <1000,
    'bg-danger': pPrice >5000 && pPrice <500000,
}" >
    Ahmed Mostafa
</div>

```

Directive ngStyle

```

<div [ngStyle]="{
    'background-color': 'red',
}"
    Ahmed Mostafa
</div>

```

Share data between 2 components

Send data from parent to child

لو انا عايز اشير داتا ما بين 2 component بس دا مش هينفع غير فى حالة واحدة لو انا عندي component هو parent و component تاني child طب يعنى ايه مثلا لو انا عندي component اسمه product ودا ال parent بتاعى وعندي component تاني اسمه products ودا ال child بتاعى وانا عايز اباصي داتا موجودة فى ال parent اباصيها لل child وهي طريقة مش مستخدمة كتير بس لازم نكون عارفنها لانها سؤال انترفيو وتعال نجرب

```
// in ts product Parent add variable
userName:string='ahmed';
// in products Child add child variable to receive data in it
@Input() userNameChild:string = ''; // انبوت عبارة عن دايركتف موجود فى الانجر بيعرف
الفاريبول انه هيشيل داتا تتحط في
<!-- in parent html بقوله انت ككمبوننت اتشايلد عندك فاريبول اسمه بوزر اتشايلد شايل داتا من
-->
<app-mobile-component [userNameChild]="userName" ></app-mobile-
component>
<!-- in child html هنا بقا بنادي الداتا اللى شايلها فاريبول الاتشايلد وبس كدا
-->
<h1> {{userNameChild}}</h1>
```

Send data from child to parent

لو انا عايز اخذ داتا من ال child اوديها الى ال parent ,
تم فهم الفكرة وسيتم اضافتها لاحقا

Services

بتقدر من خلالها تعمل شير للداتا بين اكثر من كميوننت وبكدا احنا بنتجنب تكرار الكود
وبنضيف في السيرفس اى داتا انا عايزها تكون شير علي كل الكميوننتس بتاعتنا ونوع الداتا دى ممكن تكون function او variable او for او if او ...

Steps to Create Services

Create file service in terminal add this line -1

ng g s our مثال ng g s nameService

Create data in service file like this -2

```
// variable in class service to share it in all component
userAddress:string = "xyz";
```

Import this service in any Component like this -3

```
import { OurService } from '../our.service';
```

-4 In constructor component call this service like this constructor(private _ourService:ourService)

نوتس بالعامية : هناخد نسخة من السيرفس دي داخل القوسين بتوع الكونستراكتور بالشكل السابق .. يلا بينا نـ implement دا بقا

-5 هجيب الداتا من السيرفس ولو انا عايزها ترن اول حاجة داخل ال component هنادي علي الداتا داخل ال constructor وهعمل variable يشيل او يستقبل الداتا اللي جايه من السيرفس دي

```
// variable to receives data from service
userAddress:string = '';
// constructor to copy version from service and call data in
constructor scope
constructor(private _ourService:OurService )
{
    this.userAddress = _ourService.userAddress;
}
```

ولو انا عايز الداتا اللي جايه من السيرفس ترن لما ادوس على زرار مثلاً هاخد نسخة من السيرفس في الانستراكتور وهعمل method جوا class الكمبوننت بتاعي وانا دي علي الداتا داخل ال method دي بدلا من ال constructor ومتساش تضيف الزرار في ال html وتعمل ايفنت لل method مثال

```
// variable to receives data from service
userAddress:string = '';
// constructor to copy version from service and call data in
constructor scope
constructor(private _ourService:OurService ) {}
// method
test(){
    this.userAddress = this._ourService.userAddress;
}
<button (click)="test()" class="btn btn-success">click</button>
```

ملحوظة : اي property بعرفها جوا الكلاس لو عايز استخدمها جوا method جوا نفس الكلاس لازم اقله this.variable

طيب ليه انا استخدم طريقة private وخذت نسخه من كلاس السيرفس داخل ال constructor ومستخدمتش الطريقة بتاعت ECMAScript اللي

بعمل فاريبول من النوع اني وجوا اسكوب الكونستراكتور اخذ نسخة من السيرفس بالشكل دا

```
x:any;
constructor()
{
    this.x = new OurService();
}
```

لان دا هيسبيلي مشكلة وهي انى لو خدت نسخة من الكلاس بالشكل دا فهو اول حاجة هيروح ينادي عليها هي الكونستراكتور بتاع السيرفس طيب لو انا جيت ضفت براميتز للكونستراكتور بتاع السيرفس فيما بعد هحتاج اروح في كل كمبوننت مستخدم السيرفس دي اديها براميتز في النسخة بتاعت كلاس السيرفس ودا مش صح والصح اننا نستخدم الطريقة الللى متفقين عليها فى الاول لان هى بتقوم بعمل Dependency injection اي بتقوم بحقن الاعتمادات بتاعت التمبلت بتاعتى كلها بالكامل وبتجيبه كلاس السيرفس كله بالكامل وبقوله private وبرايقت عبارة عن حاجة اسمها access modifiers ومعناها ان اي حاجة هتستخدمها مش هتقدر تستخدمها غير جوا الكلاس فقط. على عكس public تقدر تستخدمها جوا الكلاس وبر الكلاس ف انا هنا بقوله برايقت عشان معنديش اي سيناريو اقدر استخدم السيرفس خارج الكلاس بتاع الكمبوننت وبس كدا سلام .

Design Patterns: هو تصميم نمطي لاي مشكلة وهو كونسبت موجود في اي لغة برمجة.

Explain Services

هي عبارة عن كلاس عادى جدا

اماكن تعريف الـ Service بتاعتي يوجد 3 طرق وهما

1- **اضافة الـ Service فى ملف السيرفس نفسه (على مستوى كل الابليكشن)** وبتكون داخل الـ decorator الللى اسمه @Injectable وبتكون موجودة تلقائي عند انشاء ملف الـ service وبتكون shared علي كل الابليكشن . ملحوظة اي update هيتم علي السيرفس هيتم علي اي component مستخدم السيرفس دي وعلي اي module موجود في الابليكشن .

2- **اضافة الـ Service (على مستوى الـ component فقط)** وهي اني بروح لل component الللى انا عايزه يستخدم الـ service واضيف ف الـ decorator الللى اسمه @component الللى موجود داخل ملف الـ ts بتاع الـ component كدا providers:[ourServices] وهكذا علي اي component عايز استخدم في الـ service بس خلي بالك هو كدا بياخد نسخة من السيرفس يضيفها ل component الـ home مثلا وبعد كدا بياخد نسخة يضيفها ل component الـ about مثلا ولذلك اي update هيحصل ع السيرفس الموجودة ف component ما هيسمع في الـ component دا فقط.

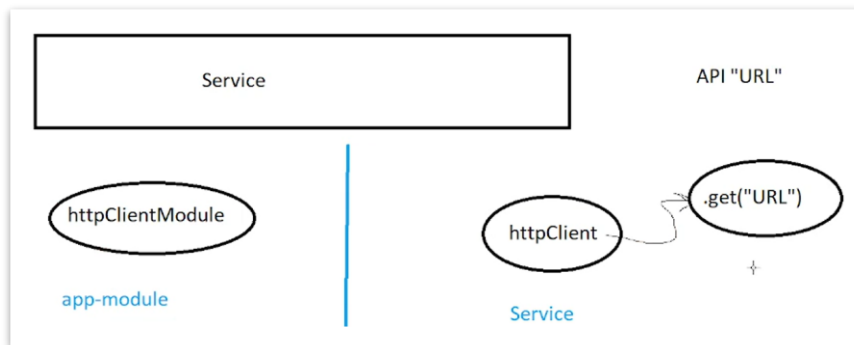
3- **اضافة الـ Service داخل الـ app.module (على مستوى الـ app.module فقط)** هتلاقي providers:[] هتضيف فيها اسم السيرفس بتاعتك . ملحوظة هنا اي update هيتم علي السيرفس هيتم علي كل الـ components الللى تحت الـ module دا.

- - ملحوظة : providers معناها بالعربي امتدد او استخدم فبقوله استخدم السيرفس دي .

Get Data from API

ندخل على مرحلة اني اجيب الداتا بتاعتي من API وليس لوكال وال API دا عبارة عن URL

وعشان اجيب Data من API لازم يكون عندي Service اضيف فيها الكلاس الللى اسمه httpClient عشان اقدر استخدم الـ method الللى اسمها get("URL") الللى من خلالها اقدر اجيب الـ Data من Api بس استنا عشان اقدر استخدم كلاس الـ httpClient لازم استخدم الـ module الللى اسمه HttpClientModule وبعمله import فى الـ app-module وهو built in فى الـ angular نازل معاها يعنى. ويلا مثال بعد الاسكرين شوت



خطوات العمل :

```
// in App-Module Import HttpClientModule
import {HttpClientModule} from "@angular/common/http";
// in Service import HttpClient class and observable
import {HttpClient} from "@angular/common/http";
import {Observable} from 'rxjs';
// in service constructor take copy from httpClient class
constructor(private _httpClient:HttpClient) { }
// create method to get data from API
getData():Observable<any>
{
  return
this._httpClient.get("https://jsonplaceholder.typicode.com/posts");
}

// in ts component call service and get data in this component
constructor(private _ourService:OurService )
{
  _ourService.getData().subscribe((data)=>{
    console.log(data)
  })
}
```

تعال نحكي قصة ونشرح يعنى ايه **Observable** طالما بقوله رجعلي داتا من api ف لازم اعمل async وعشان اعمل async لازم اعمل await و await بتيجي على حس promise لان بحول حاجة من async الى sync وبعد ال promise بعمل ال then بتاعتي ولكن دي فيها عيوب كتيرة واشهرهم ان ال promise اول ما بعرف الداتا دي من النوع promise بيروح بتنفيذ طول تلقائي فبالتالي انا محتاج حاجة تحولي من async الى sync وتجهزها ولما اعوزها اقولها اعمليلها run وفي عيب تاني وهو اني لو جايب داتا في صفحة ال home مثلا وانا روحت على صفحة ال about فال home بيفضل يعمل process مع اني مشيت من عنده ومش محتاجه في حاجة ودا مش افضل حاجة طبعاً ومحتاج حاجة تقوله لما امشي من الصفحة دي وقف ال process بتاعتها ف احنا عندنا حاجة بديلة او طريقة بديلة تنفذني من العيوب دي وهي observable وعشان استخدمها لازم اعمل import لمكتبة اسمها rxjs لان بداخلها ال observable و انا بقوم بإنشاء method داخل السيرفيس بتاعتي وبداخلها بقولها رجعلي الداتا دي من ال api و اقول ان ال method دي من النوع observable والتايب بتاعها <any> عشان الداتا

بتاعتي تكون ديناميك معايا ومن خلال ال observable اقدر اعمل subscribe او unsubscribe على ال method لما احي استخدمها في ال ts بتاع اي component عندي لانهم موجودين داخل ال observable فبالتالي اقدر اورثها وبستخدم ال subscribe عشان اشغل ال observable و ال subscribe دي بداخلها object وال object دا في 3 property وهما

- 1- Next ودي معناها انه راح جاب الداتا ورجع ورمها في parameter
- 2- Error عبارة عن فانكشن بترجعلي الايروور بتاعي لو حصل معايا ايروور واقدر اظهره ودي بديلة عن try , catch
- 3- Complete عبارة عن فانكشن بتعبر عن الخطوة التالية بعد ال next or error فهي خطوة زيادة وطريقة كتابتهم بتكون بالشكل دا كدا

```
constructor(private _ourService:OurService )
{
  _ourService.getData().subscribe({
    next: function(data){console.log(data)},
    error: function(error){console.log(error)},
    complete: function(){console.log("Al7amdullah")}
  })
}
```

ولو انا متأكد من الداتا بتاعتي وكويسة ومفهاش اخطاء ممكن استخدم الطريقة المختصرة لاستخدام ال subscribe اللي استخدمتها في المثال السابق.

component life cycle

ال component بيمر بكذا مرحلة وهما

- 1- ngOninit هي اختصار لكلمة on initialization وهي method بترن لوحدها اول ما ال component بتاعك بيتوجد او يرن فهي تشبه ال constructor لانه بيرن برضو اول ما ال component بيرن ولكن ال constructor بيرن الاول لانه اصلا حاجة موجودة اساسية داخل ال class انما ال ngOninit فهي دورة او خطوة من دورة حياة الكمبوننت وهي عبارة عن انترفيس وبترن طيب ليه هما موجودين طالما ب رأس بعض. تعالى نتكلم بميرضي الله ال constructor مبنكتبش في حاجة فهو موجود علي حس ال class واي حاجة نكتبها جوا ال ngOninit فبالتالي بعرف فقط private كذا ك parameter في ال constructor وال logic بتاعي بكتبه داخل الاسكوب بتاع ال ngOninit.
- 2- onChange ودي بترن لما يحصل اي تغير في ال component وهي عبارة عن انترفيس ف لازم اعمله import فوق واعمله implement في ال class وطريقة اضافته

```
import {Component, Input, OnChanges, SimpleChanges} from
'@angular/core'; // 1
export class ViewClientInvoicesComponentComponent implements
OnChanges{ // 2
  ngOnChanges(changes: SimpleChanges): void { //3
    console.log("hiiii");
  }
}
```

- 3- ngOnDestroy ودي معناها لما ال component بتاعي يتدمر او لما انقل على component تاني يحصل اكشن معين وطريقة اضافته

```
import {Component, Input, OnDestroy} from '@angular/core'; // 1
export class ViewClientInvoicesComponentComponent implements
OnChanges, OnDestroy{ // 2
    ngOnDestroy(): void { // 3
        console.log("hiiii");
    }
}
```

ودا ممكن استفيد منه فى انى اوقف العملية بتاعت ال Api واقوله وانت ماشي من ال component اعمل unsubscribe وخطوات العمل هي

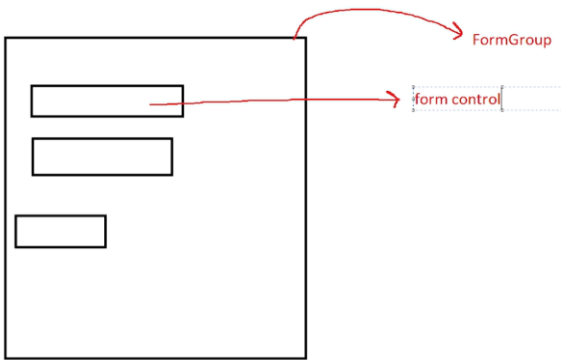
```
// create variable to copy version from ourService after get data
because service is private
mySubscribe:any;
// in ts component call service and get data in this component
constructor(private _ourService:OurService )
{
    this.mySubscribe = _ourService.getData().subscribe((data)=>{
        console.log(data)
    })
}
ngOnDestroy(): void { // to stop subscribe when toy left page
    console.log("hiiii");
    this.mySubscribe.unsubscribe();
}
```

ال URL بيكون عبارة عن حاجتين 1- path URL ودا بيكون الهوست www.facebook.com و2- هي ال Endpoint .
ال postman برنامج ب test من عليه ال API قبل ما استخدمها عشان ممكن يكون فيها خطأ او حاجة ومشكش في الكود بتاعى كـ Frontend .

Forms in Angular

الـ Angular عندها نوعين من الـ Forms وهم Reactive Forms و Template Driven Forms
Reactive Forms: بستخدمها لو ال forms بتاعتى متطلببتها كبيرة وسهلة الاستخدام وهي الاكثر استخدام
Template Driven Forms: بتفوكس بس او بتركز بس علي ال simple Scenarios وبتكون اقل في الحجم من الثانية. غير مستخدمة.

--- عشان اعمل Form عندي في ال Angular بروح استدعي module اسمه ReactiveFormsModule داخل ال App-Module وبعد كدا بعمل Import لحاجتين وهما ال FormGroup و FormControl و كمان ال Validators عشان اعمل validate على المدخلات.



FormGroup هي بمثابة الصندوق الذي بيثّل ال forms control و بعملها import في ملف ال ts بتاع ال component اللي هيكون في ال form وبعد كذا بأخد منها object عشان بداخله اعمل ال controls بتاعتي واخد من ال controls دي object وبيأخد مني optional parameter 2 وهما ال default value, validators .

Register Form

وتعال نجرب على ال

```
// in App-Module Import ReactiveFormsModule
import {ReactiveFormsModule} from '@angular/forms';
// in App-Module Import HttpClientModule
import {HttpClientModule} from "@angular/common/http";

imports: [ // بضيف اسم الموديول هنا
  HttpClientModule , ReactiveFormsModule
]

// Create auth Service and import HttpClient, Observable
import {HttpClient} from '@angular/common/http';
import { Observable } from 'rxjs';
  constructor(private _HttpClient:HttpClient) { }

  registerData(formGroupValue:object):Observable<any>{
    return this._HttpClient.post("https://route-egypt-
api.herokuapp.com/signup",formGroupValue);
  }

// in ts component Import FormGroup,FormControl, Validators
import {FormGroup,FormControl, Validators} from '@angular/forms';
  registerForm = new FormGroup({
    "first_name" : new FormControl(null,[Validators.required ,
Validators.maxLength(50)]),
    "last_name" : new FormControl(null,[Validators.required ,
Validators.maxLength(50)]),
    "email" : new FormControl(null,[Validators.required ,
Validators.email]),
    "password" : new FormControl(null,[Validators.required,
validators.pattern('^[A-Z]${}')]),
```

```

    "age" : new FormControl(null,[Validators.required ,
Validators.min(10) // or max

    }) // after that create your form in html this component
        <form (ngSubmit)="SubmitForm(registerForm)"
[formGroup]="registerForm" class="row g-4">

            <div class="col-md-6 col-sm-12 text-end">
                <label for="inputEmail4" class="form-
label">الاسم الثاني</label>
                <input formControlName="last_name" type="text"
class="form-control text-end inputsStyle shadow"
id="inputName4">
                <!-- to Handel Errors and control it -->
                <div
*ngIf="registerForm.get('last_name')?.errors &&
registerForm.get('last_name')?.touched"
class="pt-3">
                    <div
*ngIf="registerForm.get('last_name')?.getError('required')"
class="TextcolorValidation">حقل مطلوب -
</div>
                    <div
*ngIf="registerForm.get('last_name')?.getError('maxlength')"
class="TextcolorValidation">اقصي عدد
</div>
                    - احرف هو 50 برجاء التعديل
                </div>
            </div>

            <div class="col-md-6 col-sm-12 text-end">
                <label for="inputName4" class="form-label">الاسم
الاول</label>
                <input formControlName="first_name"
type="text" class="form-control text-end inputsStyle shadow"
id="inputName4">
                <div
*ngIf="registerForm.get('first_name')?.errors &&
registerForm.get('first_name')?.touched"
class="pt-3">
                    <div
*ngIf="registerForm.get('first_name')?.getError('required')"

```

```

class="TextcolorValidation">حقّل مطلوب -
</div>
    <div
*ngIf="registerForm.get('first_name')?.getError('maxlength')"
class="TextcolorValidation">اقصي عدد
    </div>
    </div>
    </div>
    <div class="text-end">
    <label for="inputEmail4" class="form-
label">البريد لاللكتروني</label>
    <input formControlName="email" type="email"
class="form-control text-end inputsStyle shadow"
id="inputEmail4">
    <div *ngIf="registerForm.get('email')?.errors
&& registerForm.get('email')?.touched"
class="pt-3">
    <div
*ngIf="registerForm.get('email')?.getError('required')"
class="TextcolorValidation">حقّل
    </div>
    <div
*ngIf="registerForm.get('email')?.getError('email')"
class="TextcolorValidation">برجاء
    </div>
    </div>
    </div>
    <div class="text-end">
    <label for="inputPassword4" class="form-
label">كلمة المرور</label>
    <input formControlName="password"
type="password"
class="form-control text-end inputsStyle
shadow" id="inputPassword4">
    <div
*ngIf="registerForm.get('password')?.errors &&
registerForm.get('password')?.touched"
class="pt-3">

```

```

        <div
*ngIf="registerForm.get('password')?.getError('required')"
        class="TextcolorValidation">حقن مطلوب -
</div>

        <!-- <p
*ngIf="registerForm.get('password')?.getError('maxlength')"
class="TextcolorValidation">maxLength is 50</p> -->
        </div>
    </div>

    <div class="text-end">
        <label for="inputAge" class="form-
label">السن</label>
        <input formControlName="age" type="number"
class="form-control text-end inputsStyle shadow"
        id="inputAge">

        <div *ngIf="registerForm.get('age')?.errors &&
registerForm.get('age')?.touched" class="pt-3">
            <div
*ngIf="registerForm.get('age')?.getError('required')"
class="TextcolorValidation">حقن
                مطلوب -</div>
            <div
*ngIf="registerForm.get('age')?.getError('min')"
class="TextcolorValidation">maxLength
                is 10</div>
            <div
*ngIf="registerForm.get('age')?.getError('max')"
class="TextcolorValidation">maxLength
                is 60</div>
            </div>
        </div>
        <!-- create button submit and disable it if form
is invalid -->
        <div class="mt-4">
            <button [disabled]="registerForm.invalid"
type="submit" class="w-100 shadow btnStyle">تسجيل</button>
        </div>

    </form>

```

```

        <!-- to handel error if form return error will show
here -->
        <div *ngIf="registerErrorMessage"
class="TextcolorValidation">{{registerErrorMessage}}</div>

```

Programing Routing

من خلالها اقدر اروح على صفحة ما من خلال navigate الموجودة داخل كلاس اسمه Router ودا لو عايز اقله لما تنفذ method دي افتح الصفحة دي

```

// in ts component import Router
import { Router } from '@angular/router';
// take copy from router class in constructor
constructor(private _Router:Router){}
// create method to route in home page
navigatorMethod(){
    this._Router.navigate(['home']);
}

<!-- in html Create btn -->
<button (click)="navigatorMethod()" class="btn btn-info my-5">navigate</button>

```

Jwt Library, Token

ال Token بيكون شاييل في الداتا بتاعت ال user اللي عامل ال login دلوقتي ولازم تكون متشفرة encoding ف انا بفك التشفير دا وبعمله حاجة اسمها decode وبعد كدا احطه في ال local storage عشان اقدر استخدم الداتا بتاعت اليوزر دا استخدمها في اكثر من مكان مثلا في ال home او ال about .

مكتبة JWT (Json web token) هي مكتبة بتساعدني في اني اعمل Encoded او Decoded للداتا بتاعتك. وهي مكتبة موجودة في ال Angular. فيروح على ال terminal بروح انزلها `npm i jwt-decode` وادوس انتر وعشان استخدمها بروح اعملها import في ال service بتاعتى عشان تكون shared لاي component بالشكل دا

```
import jwt_decode from 'jwt-decode';
```

وبعد كدا بعمل method وبعمل variable من النوع any استقبل في الداتا بعد ما يتعملها decode وبعمله عن طريق method اسمها jwt_decode() وبتاخذ parameter من خلاله بجيب ال token من ال local storage وبقله JSON.stringify عشان هي لا تقبل ال null او ان ال token يكون فاضي فبقوله لو الداتا null حولها ل string وبتكون بالشكل دا كدا

```

userData : any = null; // create var
decodeUserData(){

```



```

    this.userData =
jwt_decode(JSON.stringify(localStorage.getItem("userToken")))
}

```

دي طريقة فك ال token اللى بيكون جاي من ال backend لما بعمل login تعال بقا نوظف دا وازاي نستخدمه واحنا بنعمل login code

Login Form

```

// in App-Module Import ReactiveFormsModule
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
// in App-Module Import HttpClientModule
import { HttpClientModule } from "@angular/common/http";
imports: [ // بضيف اسم الموديول هنا
    HttpClientModule, ReactiveFormsModule
] // بعد كدا اروح على السيرفيس بتاعتي اجهزها
// in my service import HttpClient, Observable, jwt_decode
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';
import jwt_decode from 'jwt-decode';
userData : any; // فاربيول بستيقل في الداتا بعد ما يتعملها ديكود او فك التشفير

constructor(private _HttpClient:HttpClient) { }

loginData(formGroupValue:object):Observable<any>{
    return this._HttpClient.post("https://route-egypt-
api.herokuapp.com/signin",formGroupValue);
}
// create method to decode Token (decode userData)
decodeUserData(){
    this.userData =
jwt_decode(JSON.stringify(localStorage.getItem("userToken")))
} // بعد كدا اروح على التايب اسكريبت بتاع الكمبيوتر بتاعتي اجهزه
// in ts Component import this
import { FormGroup, FormControl, Validators } from '@angular/forms'
import { AuthService } from '../auth.service';
import { Router } from '@angular/router';

loginErrorMessage : string = '';

loginForm = new FormGroup({
    "email" : new FormControl(null, [Validators.required ,
Validators.email]),

```

```

    "password" : new FormControl(null, [Validators.required]),
  })

  constructor(private _AuthService:AuthService ,private
_Router:Router){}

  loginMethod(_formGroup:FormGroup){
this._AuthService.loginData(_formGroup.value).subscribe((response)=>{
  if(response.message == 'success'){
    localStorage.setItem("userToken",response.token); //
    this._AuthService.decodeUserData();
    this._Router.navigate(["home"]);
  }
  else{
    this.loginErrorMessage = response.message
  }
})
}

```

BehaviorSubject navbar

هنبداً بقا نضبط ال navbar ونضبط ال security بتاعتنا وهكذا يلا بينا