

السبت

Saturday

5

٢٠١٥

٢٣ صفر ١٤٣٧ هـ

٢٥ هاتور ١٧٢٢ ق

ديسمبر



by Google

* client side framework go angular
هو يعتمد على application framework
من الـ client-side

* خبرة اسهامات ١٥٪ تكون مأهولة قبل ما تبدأ

① need to know TS
② need to know OOP
③ design patterns

السبت الأحد الاثنين الثلاثاء الأربعاء الخميس الجمعة				
٢٣ ٢٤ ٢٥ ٢٦ ٢٧ ٢٨ ٢٩	٢٠ ٢١ ٢٢ ٢٣ ٢٤ ٢٥ ٢٦	١٨ ١٩ ٢٠ ٢١ ٢٢ ٢٣ ٢٤	١٥ ١٦ ١٧ ١٨ ١٩ ٢٠ ٢١	١٢ ١٣ ١٤ ١٥ ١٦ ١٧ ١٨

December

٤

4

الجمعة

Friday

2015

ديسمبر

٢٤ هاتور ١٤٣٧

٢٢ صفر ١٤٣٧

intro

Session 1

- 1- angular intro
- 2- angular setup
- 3- angular files
- 4- SPA: technique / concept
- 5- angular architecture
- 6- project flow
- 7- install bootstrap & fontawesome

12

1

Sunday, December 27, 2015

2

Tuesday, December 29, 2015

zainab, naseeb

3

4

5

6

December

Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

① Angular Intro

ما هو ال angular

ال development framework هو angular
ال single page application ، المبني على TypeScript
و يستعمل libraries ل التعامل مع API
..... testing وال validation وال DI و وال

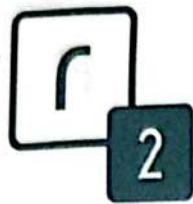
ما الفرق بين angular2 و angularjs

angularjs ظهر من ٢٠١٤ لل js
، لكنه كان مختلفاً تماماً الاختلاف عن angularjs
كان على غير حرم كامل لل js و بناء framework جديداً
، تماماً باسم Angular2 .
ال js مبني بال ts .
angularjs مبني بال js و لـ performance better من angular2

ما هو ال @angular/cli في مرحلة ال development

ال tool لـ angular cli
- command shell لـ angular applications
- command line interface

ما هو ال angular routing
ال routing هو التنقل بين الصفحات / تغيير ال URL path
الـ SPA عن طريق ال



② Angular setup

- * pre-requisites?
 - ① HTML, CSS, JS, TS long term support
 - ② Node.js / npm LTS

٩. إدخال المكتبات الضرورية في المجلد

١٠. * install the Angular CLI in your machine.

١١. * > npm i -g @angular/cli

١٢. * install global (-g) مكتوى بـ (C: \) كل يعنى أعمل من الـ

١٣. * create angular project:

١- create a workspace by using CLI

٢. > ng new app1

٣. > cd app1

٤. > ng serve --open → run local server

٥. ٤- create components

٦. > ng g c navbar

٧. ٤- > npm i package-name مع الـ dependencies

٨. > npm i pack-name --save-dev

٩. ٤- حفظ dependencies مع الـ devDependencies

١٠. ٤- حفظ dependencies من مرحلة development

١١. ٤- deployment / production

١٢. ٤- > npm i لـ package.json من الـ

Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

③ angular files

• node dependencies \rightarrow كل الـ node-modules \rightarrow npm \rightarrow install \rightarrow all packages \rightarrow دينية او الـ packages من المترقب.

• source files \rightarrow src \rightarrow مفهارة من الـ components \rightarrow يحتوى على

components \rightarrow يحتوى على كل الـ app folder \rightarrow وكل الداتا وكل الـ app logic

static files \rightarrow يحتوى على assets \rightarrow

نوع الصور والـ CSS وـ icons والـ JS

icon \rightarrow favicon.ico \rightarrow يمثل tab

جنب الـ

entry point \rightarrow index.html \rightarrow

main html file

files path \rightarrow يمثل gitignore \rightarrow path

track \rightarrow متى يتم إغفال folders

version control \rightarrow متى يتم إغفال

all packages \rightarrow يحتوى على اسماء الـ package.json

الـ packages يستخدمه والـ

الـ development يستخدمه.

وـ production

reference \rightarrow دالة الـ

يتأتى دالة ملف

الـ requirements

نوفمبر

٢٠١٧٢٢ هـ

١٤٣٧ هـ

④ SPA technique

٨. قبل الـ SPA concept مقدمة

٩. HTTP request ينجز بوجهة HTML page او اى

HTTP response من page دير وار sever

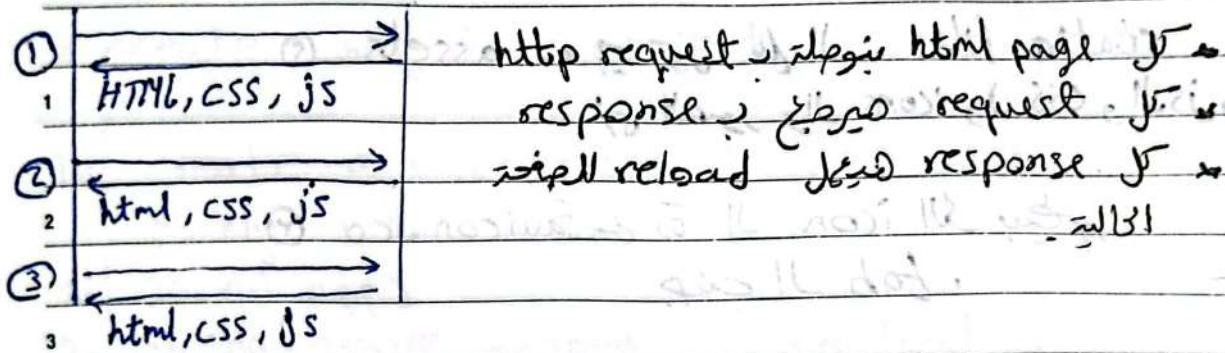
١٠. باردة من ملف ال HTML وال اللغات الـ CSS الـ JS

١١. يحمل Browser الملفات من المكان لازم

١٢. reload page او refresh

المفهوم / الـ page الجيد .

client server



٤. اما كل العالية بين الـ request والـ response بتاخذ وقت بالثانويدة .

٥. refresh على الـ performance على الـ

٦. مستاجر على الـ user experience

٧. هو الـ SPA يتعامل مع كل ملفات الـ HTML ملف html واحد .

٨. الـ SPA technique / concept هو ذكرى انه ازاي نقل

٩. عدد الـ requests خدر الامكان بالثانويدة .

١٠. الـ refresh الـ بجهة وده عن طريق ان احنا ندخل الى

١١. request كل صفحات الـ HTML مرة واحدة مع اول

November

Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

⑤ Angular Arch.

الangular يعتمد على component-based-arch. وهذا يعني أن الوحدة الأساسية/main unit لها متغير، مبنى بالangular هو ال component.

يُمكن عَكَن نقول إن ال angular مبنية من مجموعة components ذات حاجات ثانية تخدم ال components دفعاً.

أيّ هو ال component

الangular component هو جزء معين من مشروع ال angular، والجزء الذي يكون scalable يعني جزء من المشروع يمكن بيعه أو تكراره.

ال component يكون reusuable يعني أقدر أعيده واستخدامه أكثر من مرة في المشروع عن طريق ال selector.

كل component من ال angular يتكون من 3 files:

① ملف ال html

② ملف ال CSS

③ ملف ال ts

كل selector يحتوى على component خاص بيَعِدُونه عبارة عن tag (يُعنِّي custom html tag) له اسم selector من طريق ال <selector> نقدر نوصل المحتوى.

ال selector يسْتَغْلِل وسيحْتَاط مكانة ال html بناءً على selector.

(جِئْجِيْه ال call باسم ال function)

نوفمبر

١٨ هاتور ١٤٢٢ في

١٤٣٧ هـ صفر

٧ اجزء هو ال app component

٨ parent component او ال root component دو ال
الك يحتوى على كل ال components (باتقة المترافق)

٩ عن by default يتم انشائه ال app component
١٠ طريق ال طريق . angular

١١ انت هو ال app component بال Selector
١٢ يكتب في ملف ال index.html و عن طريق نوصول
كل ال components

١٣ NgModule هو ال app module

١٤ component هو مجموعه من ال module
١٥ module يقع component لازم ال

١٦ default module هو ال app.module.ts ال ينزل

١٧ مع ال component لازم منتفق ال Angular فنى ال

١٨ declaration يتابع ال app module

١٩ بالذات ذكرنا خدم selector

٢٠

٢١ class هو طبارة عن angular component
٢٢ component عن class يميز ال
٢٣ @Component (in, decorator function) مفهوم



app component

```
→ app.component.html → <app-nav></app-nav>
app.component.css
app.component.ts
<app-root></app-root>
```

nav component

```
nav.component.html
nav.component.css
nav.component.ts
<app-nav>
```

home component

```
home.component.html
home.component.css
home.component.ts
<app-home>
```

index.html

```
<app-root></app-root>
```

NgModule

الـ modules بسيطة هو أي ملف يكتفى
عاليز حشر كافـ يشتـوا بـ بعضـ فـ يـ عـ الـ module



26

الذبيش

Thursday

2015

نوفمبر

١٧٣٢ هـ

١٤٣٧ مـ

⑥ project flow

typescript class ما هو الـ "component" class يتصارع مع الـ "decorator" class عن طريق component decorator يكتسب الميزات التي تختلف عن angular component في fact module يتم استيراده من "Component" في angular/core "@angular/core"

عن طريق الـ @Component decorator يكتسب الـ "Component" class الميزات المادية المائية التي يكتسبها عن طريق metadata يحتوي على الـ "Component" metadata يكتسبها عن طريق selector اسم الـ "Component" selector يمكن ان يكون html file مكان او css file مكان

ex)

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

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

```
export class HomeComponent {
```

angular module لدوم نفتح جوا angular component ولدينا selector الـ "app-home" بتاعة

November

Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16



recap

١- الـ component-based-arch قائم على فكرة الـ angular وده مفهوم أن نبتاع صفحات مننا حابطة اسمها page لكن عن طريق اسماً component

٢- كل component يتكون من :

html file ①

css file ②

ts file ③

٣- نرسم أي selector لـ component أو selector يوصلنا إلى ملائكة ثالث .
Default component يكون معاً angular by default . app component

٤- كل selector أو component (ng ٤ c home) ← angular/cli configuration files كل files وال configuration files .
الامر ده يحول selector إلى class معناته هو الـ app module وحيط selector class

٥- الـ components container هو الـ angular module .
angular module يحتوي على selector component

٦- ملف الـ index.html يحتوى على كل selector .
الـ app component يحتوى على كل components .
متاحة الـ project ، مش منفرد من الـ index file خالص .
كل صاحب Selector له html file متاح الـ app component

- 7 - ملف package.json هو أقدم ملف من المشروع
- 8 - لبيان أطوال روت مبارزة عن الـ history الذي عرفه
- 9 - انتقاله إلى npm install ، الملف هو سيكون فيه `install` الـ packages
- 10 - اسم الـ package
- 11 - يشير إلى كل package بتابع كل version
- 12 - لبيان نقل الـ packages إلى install
- 13 - npm يشير إلى package.json
- 14 - الـ module هو مبارزة عن `angular` مع بعده `components`
- 15 - behavior يشير function أو decorator
- 16 - class أو metadata
- 17 - يستخدم الـ decorator لبيان معلومات
- 18 - عن الـ component أو عن الـ file التي فيه
- 19 - `@Component()` هو الـ component باز الـ decorator
- 20 - `@NgModule()` هو الـ module
- 21 - ملف الـ index.html من هنا
- 22 - حفظ الـ 2-terminals يمكن رفع VS code
- 23 - الأول run في الـ server
- 24 - الثاني run في الـ cli من جامد ثانية

٧ angular.json

الـ configuration هو ملف يُستخدم في angular.json
الخاص بالـ angular زى مثلاً :

١ او create يستعمل من طرق الـ cli
" يُكتب قبل اسماء app-

٢ ملفات الـ .css الخارجية زى الـ bootstrap.css
٣ ملفات الـ js الخارجية زى الـ jquery.js

اى تغيير في ملف الـ angular.json ، عثمان يسْتَعِجِّل
الازم نقل الـ server ونستله ثان .

٤ جوا ملف الـ angular.json فيه property :
relative path (array) توضع فيه الـ paths
اـ اسماء (array) تكتب فيه الـ files
، متاع ملف الـ css الخارجى .

٥ اـ اسماء (array) تكتب فيه الـ files
متاع ملف الـ js الخارجى .

٦ ملف الـ configurations في angular.json
يسْتَعِجِّل على كل الـ angular components ، عثمان
كده اهنا بربط ملفات الـ .css والـ .js الخارجية من ملف
الـ angular.json بحيث يسْتَعِجِّل من كل لـ components .



22

الأحد

Sunday

20

نوفمبر

١٢ هـ ٢٠٢٢ ق

١٤٣٧ هـ

١) (ex) How to install bootstrap?

1-) >> npm i bootstrap

2-) in angular.json file:

```

{
  "styles": [
    "./node_modules/bootstrap/dist/css/bootstrap.css"
  ]
}
  
```

```

  ],
  "scripts": [
    "./node_modules/bootstrap/dist/js/bootstrap.js"
  ]
}
  
```

```

  ]
}
  
```

3-) ادخل ال serve واعمل run ثانية

عن التغييرات تتحقق فوراً وتتبع

4-) كل ال components

>> Ctrl + C

5-) >> ng serve -o

6-) **ملاحظة**: هو ملف ال angular.json فيه أكثر من "scripts" و "styles" ، إنما إنما يتابع ال path 'css' أو ال 'js' في "scripts" & "styles" property أول

November

Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30																		

٧) ما هو Webpack

هي أداة بدل packaging لملفات html وال CSS

والمفهوم يشير إلى إنشاء ملف html من المجموع

من ملف html واحد وكل ملفات ال js هي ملف واحد

وكل ملفات ال js css من ملف CSS واحد مع مراعاة ترتيب

الخطوات

٨) دیکوپ های اولیه

والمفهوم يشير إلى إنشاء ملف json يوضح المجموعات

وامتداد الملفات الكائن في دليل (angular.json)

٩) ال module يربط ال components مع بعضها

ملف ال ts هو رئيس جلس لدارة ال component



↳ ng new pl --style=scss ↳

↳ cd pl

↳ ng add @angular/material

٢٠

الجمعة

20

Friday

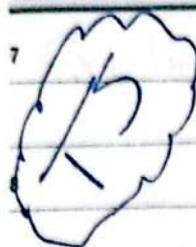
نوفمبر

١٧٣٢ هـ

١٠ هـ

٢٠

١٤٣٧ هـ



session ③

- ٩ ١- typescript intro
- ٢- typescript datatypes
- ١٠ ٣- generic types
- ٤- interface , class , type
- ٥- enum
- ٦- oop
- ١٢ ٧- decorator
- ٨- tsconfig

١٣ ٩- tsconfig.json

٢

tsconfig.json in our project

٣

tsconfig.json in base

٤

tsconfig.json in lib folder (the project)

٥

٦

November

Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon							
١	٢	٣	٤	٥	٦	٧	٨	٩	١٠	١١	١٢	١٣	١٤	١٥	١٦	١٧	١٨	١٩	٢٠	٢١	٢٢	٢٣	٢٤	٢٥	٢٦	٢٧	٢٨	٢٩	٣٠

① TypeScript intro * ts, is, js + more.

١- الـ typescript هو لغة برمجة مبنية على لغة الـ JavaScript.
الـ ts ما هي إلا الـ js مع ميزات زادت.
ts = js + typed + interface

٢- الـ ts هو strongly typed يعني لازم نحدد نوع الـ variable

٣- الـ ts مستغل في الـ ts بالذات لازم نتحول الـ ts إلى js لأن الـ browser يفهمه بورقة عن طريق transpiles تحول الـ ts إلى js.

٤- أية ميزة الـ typescript ؟

٥- الـ ts فيحسن كثافة اللور عن طريق إيه:
① بيعظّر الـ error في مرحلة الـ development يمكن

قبل مرحلة الـ runtime على تنس الـ dz.

② بيخلاك اللور بيجمع predictive يعني متوقع نسيجية إيه.

٦- الـ ts مثل الـ dz أقرب للغافر البرمجة الثانية.

٧- بن تخشم الـ dz في مرحلة الـ development فقط، لكن من مرحلة الـ production لازم نتحول كل من طريق الـ transpiles

» npm i -g typescript ←

» tsc test.ts ← → حول لـ dz فقط

» tsc test.ts -watch ← → حول لـ dz وتابع أي تغير

نوفمبر

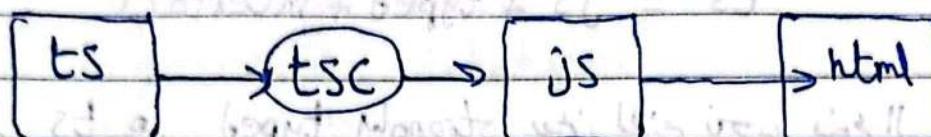
هـ ١٤٣٧ ١٧٢٢ ق

١٤٣٧ هـ

- ٧- لغة ts هي development tool يعني ما ينزل إلى developer فقط ، لكن هي مستخدمة في production.

٩-

١٠-



- ١١- الـ ts عمل لغة الـ ts التي ت歸.microsoft .
يمكنك بنفسك عمل لغة ts # بالذات لغة ts
قريبة من ts # ، يمكنك بذلك العمل على .net .
سيعملوا به خدمة ts مع angular

١-

٢-

٣-

٤-

٥-

٦-

٧-

٨-

٩-

١٠-

١١-

١٢-

١٣-

١٤-

١٥-

١٦-

١٧-

١٨-

١٩-

٢٠-

٢١-

٢٢-

٢٣-

٢٤-

٢٥-

٢٦-

٢٧-

٢٨-

٢٩-

٣٠-



الـ ts إنها من أجمل الأشياء التي يحبها developer .
متى يتحقق تصور الذي وظيفتك
الـ runtime error قبل أن

تصير مزعجة للـ ts انه typed .

الـ ts طور كسبيل للـ ts لتحسين كتابة ts وجعلها
متحفظة للـ ts .

November

Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

② typescript datatypes :

`let name: datatype = value ;`

① `let x: string = 'hello' ;`

② `let x: number = 22 ;`

③ `let x: boolean = true ;`

④ `let x: string[] = ['hi', 'hello'] ;`

⑤ `let x: object = {} ;`

⑥ tuple type → mix b/w types , الترتب

(ex) `let x: [string, number] ;`

`x = ["tashipa", 159] ;`

⑦ let x: any ; → ينتهي في حالة إى من

عارف او لا الاجابة data

⑧ union type → may be type1 or type2 or ...

(ex) `let x: string | null | undefined ;`

undefined أو null أو string يمكن تكون

(ex) `let x: string[] | number[] ;`



mixed array

(ex) `let x: (string | number)[] ;`

`x = ['hi', 15, 20, 'llb'] ;`

`let x: number[] ; ↔ let x: Array<number> ;`

الاثنين

16

الاثنين

Monday

2015

نوفمبر

١٧٢٢ هـ

١٤٣٧ مـ

③ generic types \rightarrow علامة

→ function test(x:number):void {

}

→ function test(x:string):number {

}

. type يعنى اننا نقدر لغير parameter \rightarrow

. ذكرنا خبر نوع لا ال data function \rightarrow

1

يعنى حاجة عامة

function \rightarrow من طريق generic type زعتر نوع

الا الم تدخل عليه type

3

(3) function test<hambozo>(x:hambozo[]):hambozo[] {

4

return \rightarrow الم تدخل عليه type

5

test<string>(['A', 'b', 'c']);

6

test<number>([1, 2, 3, 11]);

\rightarrow جرى التعرف ان اصنافهم generic type

call different type \rightarrow different generic type

الذان

November

Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

④ Interface

• object كلّه؟ لو كانا بالذات array of object property ممكن يكون في الـ array ممكّن مختلف عن الثاني، وكل property مختلف عن الثاني، ياخده datatype مختلف عن الثاني.

• property يمكن منحصر في كل property من الـ object والـ datatype ياخده كل property

• interface الحال فيه هو الـ interface

• الحال فيه هو الـ interface

• الـ interface هو something like custom typing هو interface object معين على الـ structure نعرض أو نأخذ معين على الـ

• الـ interface هو something like typescript something

• تحدّي بنية الـ object يعني نقرّر بنية الـ object لازم تكون من الـ object ممكّن ~~~~

• property هو الـ datatype ياخده كل property

• method هو الـ function ياخده كل method

• return هو الـ return ياخده كل return

• custom type أو class من interface أو

١٤

السبت

١٤

Saturday

٢٠١٥

نوفمبر

٤ هاتور ١٤٣٢ق

١٤٣٧هـ

أولاً نعمل interface

①

interface Product {

name: string;

price: number;

onSale: boolean;

id?: number;

↳ optional

buy(): string;

}

12

* لو نفس الاسم ، مثل فيحصل

1 inheritance و لكن ~~جاء~~ overriding

شُورت من الـ interface واخر نفس الاسم .

2

Interface Product { → interface

3 category?: string الـ فوقيـة و يـور عـلـيـه

is up property

4 } ↳ constructor a lot , it's important ... category .

let x: Product[] = [

5 { name: 'TV' , price: 600 , onSale: true ,

buy: function() {

6 return "hi"; }

7 }]

* الـ object no class ، مثل مـعـرـجـهـ أـلـيـهـ interface

datatype يـعنـ بـسـ جـمـعـ اـبـنـ custom type

↑ جـمـعـهـ جـمـعـهـ .

November

Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

(2)

```
class Product {
    name: string;
    price: number;
    onSale: boolean;
    id?: number;
    buy(): string;
}
```

②) ایه هو الغری ایه اغل بایخنیا ال class
" واری اغل بایخنیا ال interface " واری اغل بایخنیا ال custom type

الغری الجوهري ایه ال class دیتتحول ل
می ال دیز و میش دیکون لنه ایخنام من ال دیز لانه
میکول ج دیکون من ال ts من آلت.

لکن ال interface من دیتتحول لئر حاصحة من ال دیز
لئر هو سبیر custom type

الفکری من ال custom type ایه احنا نعمل
نحد بیة سکرر ال object / object structure
می مرحلة ال development . لکن من مرحلة ال production
می نتحول ملحوظ ال ts ل js ال object چوچیزی ما کان
یعنی ما چوچیز structure

(3)

```
type Product = {
    name: string,
    price: number,
    onSale: boolean,
    id?: number
}
```

ما هو نوعي الفرق بين type وال interface

إذا كان مقررنا أن type ي繼承 من

أو أن interface ي繼承 من ما زال؟

1- كم كل type جيد باسم مختلف

2- نعم أو لا ي繼承 من type إلا طبقاً لورثة منه property and &

```
type Product2 = Product & {
```

```
category: string,
```

```
}
```

* الاختلاف بين الـ type والـ interface
طريقة الـ inherit

٧. أيهما هو الفرق بين class وال interface وال type؟

١- class من ال compile يتحوال ل function.

٢- interface غير ال compile مستحول لـ class.

٣- class أو type دون فرق ال لكن طريقة inherit مختلفة.

⑤ enum

ال enum هو class أو object مثل . collection نوع من انواع ال

ال enum يحتوى على list من values محددة user يقدر اختيار منها.

(ex) enum Color {

Red, 1/0 } numeric بغير enum ال bydefault values بالترتيب للقيم الـ 0, 1, 2.

Green, 1/1

Blue, 1/2 } .

let bgColor:Color; Color[0] .
bgColor = Color.Red; \leftrightarrow bgColor = 0;

* ال datatype هو enum بـ custom typing نعمل .
مثلاً يقبل قيم محددة.

١

10

الثلاثاء

Tuesday

٢٠١٩

جنوب

باب ١٧٢٢ ق

١٤٣٧ هـ

٦ OOP in TS:

٨ object وال classes يتباع ال syntax ال features مع ES6 syntax ال members وال access modifiers إن فيه feature اوصى class اور

٩ (ex)

```
class Car {
```

```
 10 private chassisNo:number;
 11 protected motorNo:number;
 12 public wheel:number;
 13 constructor(wheel:number) {
 14   this.wheel = wheel;
 15 }
 16 disp():void {
```

١٧ let obj = new Car(15);

١٨ by default public properties كلها خارج class property ال access يمكن ذكره public من غيره private كلها عن طريق method لكن ذكره داخل class

١٩ property ال access readonly كلها منحصر في class

November

Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

الـ interface هي ساختة لا structure ولا المفروض
أو class أو object يبعض عليه.

ـ الـ function يندرج فيه الـ interface
object ، فقط بدون implementation ، والـ body
أو الـ class

ـ الـ class يقترب منه من class وله فقط لكن
ـ " يقترب أكثر من implement interface

interface I1 { }

interface I2 { }

class C1 { }

class C2 extends C1 implements I1, I2 { }

ـ الـ interface يقترب من أكثر من

interface I3 extends I1, I2 { }

ـ الفرق بين الـ namespaces والـ modules

ـ الاثنين الهدف منه هو تنظيم الـ program عن طريق
ـ تقسم الكود على أكثر من file وكل file يختص بجهاز معين

ـ الفرق إن الـ files لا متصلة في الـ namespace
ـ كل file مستمدوا من dz file واحد

⑦ decorators

- ٨. **design pattern** من **decorator** يعيننا طريقة ان اجتاز / attach حماقة من class او من ال function لكن من ال runtime يعني من تغير ما يغير حماقة من class او من ال function

10

٩. **special kind of declarations** **decorator** ال

- ١١. او method او class مع attach نزيله / **parameter** او **property**

12

١٢. **الهدف** من ال **decorator** انه يضيف

- ١٣. او method او class او ال behaviour

2

١٤. **فهي أكثر من نوع ال** **decorators** **class decorator** \hookrightarrow او

\hookrightarrow function \hookrightarrow function \hookrightarrow ال

١٥. **declaration** او **decorator** دو عبارتين عن

١٦. **method** قبل ال **class** او قبل ال **method** او

١٧. **class** يضيف ذياته على ال **functionality**

١٨. او على ال **method** او

6

١٩. **class** في ال **angular** (مثال)

٢٠. يتصرف ك **component** يضيف قبل

٢١. **@Component()** **decorator** على class

⑧ tsconfig.json

" حِر المَارِسِيُّ الْكَبِيرِ هِيَ كُوْنَ فِيْهِ كَثِيرٌ جَهَاْ " .
وَكَلُوهُ لِلَّازِمِ هِيَ تَحْوِلُواْ لِكَفِيلِ، وَهُوَ سَكَلٌ مَعْدُ جَنُورَةً .

" الْأَخْدُلْ إِنْ أَحْنَا نَغْفِلْ مَلَفَاتِ الْTٍSِ فِيْهِ فَوْهَمْ " .
" اسْتِرَ (SٍC) وَرَعِيْهِ الْFٍOٍLٍDٍEِRِ الَّذِيْ يَحْتَوِيْ عَلَىْ كُلِّ الْFٍIٍLٍEِSِ الَّذِيْ مُنْتَهِيَّهُ دَسْتِرْلَهُ دَيْلَوْ . " .
" وَنَجْوَطْ كُلِّ مَلَفَاتِ الْDٍZِ / الْMٍLٍFٍAٍTٍSِ الَّذِيْ مُنْتَهِيَّلُ " .
" (PٍUٍBٍLٍIٍC) FٍOٍLٍDٍEِRِ دَيْلَوْ " .

" إِزَى مَا نَشَلْ مَكْبِلْ مَلَفَاتِ الْTٍSِ جَوَاهِرْ " .
" إِزَى تَحْكِيْ مَلَفَاتِ الْDٍZِ تَسْبِيْتْ جَوَاهِرْ " .

" دَى عن طَرِيقِ إِنْ أَحْنَا نَسْلِلْ مِنْ الْFٍOٍLٍDٍEِRِ الْأَخْدُلْ " .
" بَالْTٍSِ ، بَالْTٍSِ ، أَدَلَّ لِلَّازِمِ تَسْتَرِيْنِ الْFٍOٍLٍDٍEِRِ الْأَخْدُلْ " .
" بَالْTٍSِ --init -- تَسْرِيْيِيْلِ الْTٍSِ " .

1) tsc --init ↪

2) open the tsconfig.json file →

3) { "compilerOptions": {

 "rootDir": ". /src",

 "outDir": ". /public",

}

4) tsc ↪] → دَيْخِسْ جَوَاهِرْ الْFٍOٍLٍDٍEِRِ وَيَحْوِلْ →
tsc -w ↪] كل الـMٍLٍFٍAٍTٍSِ لِكَفِيلِ ts file وَيَبْعَثْ
الـoutDir

١ How to generate interface with cli?

٢) `ng g i user`

٣) `user.ts`
export interface User {
}

٤) `import {User} from './user'`

لقد نعمل ملف Interface

* لاتجاه افضل كنقطة وركيزة reversability
أقررت على import واصدار خدمة من الـ component
ولو كانت تعدل معها مع كل ما

٥) يفضل نقل اسخنار الـ object type وسائله من خارج
`interface`

٦

٧

٨

٩

November

Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun
١	٢	٣	٤	٥	٦	٧	٨	٩	١٠	١١	١٢	١٣	١٤	١٥	١٦	١٧	١٨	١٩	٢٠	٢١	٢٢

٤

الأربعاء

4

Wednesday

نوفمبر

٢٤ باب ١٧٢٢ ق

١٤٣٧ هـ سبتمبر

٢٠٢١

routing
intro

Session ③

Binding
Directives
pipes

- ٨ ١ binding
- ٩ ٢ directives
- ١٠ ٣ revision
- ٣ angular routing

١١

١٢

١

٢

٣

٤

٥

٦

١

binding

(على طبق كورس ts بكتور ال html)

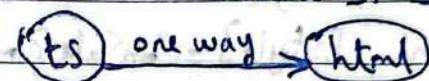
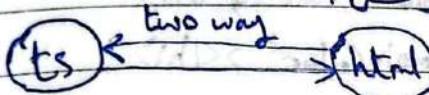
* angular يحتوي على feature هو binding
 .html وهو عبارة عن عملية ربط بين ts وال html

* ال binding ينقسم لنوعين:

two way data binding ②

one way data binding ①

"هـ عبارة عن ال مول (binding) من html الى ts من data الى ts من ts الى html"
 من ارسال data الى ts من ts الى html
 والعكس.



One way binding {{}}, [], ()

(1) data interpolation {{variable name in ts}}

(2) property binding [attribute] = "variable name in ts"

(3) event binding (event name) = "method name ()"

(ex)

```

class HomeComponent {
  x:number = 20;
  w:number = 100;
  n:string = 'Hello';
  m(e:Event){
    }
}
  
```

```

<h1>{{n}}</h1>
<img [width] = "w" />
<button (click) = "m($event)" />
  
```



الاثنين
Monday

2

2015

نوفمبر

٢٢ باب ١٧٢٢ ق

١٤٣٧ هـ حرم ٢.

٩. style binding & class.. binding (property binding)

① style binding

٩. class HomeComponent {

mainColor = 'red';

onSale = true;

mainStyle = 'color:red; font-size:30px' ;

11. } .

`<h1 [style.color] = "mainColor"></h1>`

12. `<h1 [style.color] = "onSale? mainColor:'blue'"></h1>`

`<h1 [style] = "onSale? mainStyle:mainColor"></h1>`

1

② class binding

`<h1 [class] = "onSale? 'bg-danger': 'w-100'"></h1>`

• two way data binding

① import FormsModule

② [(ngModel)] = 'name'

٦. ملخص الـ app.module.ts

١. import components الـ

built-in modules. الـ import

الـ components مع الـ

November

Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

الأحد

Sunday

1

٢٠١٥

١٤٣٧ هـ محرم ١٩

٢١ باب ١٧٢٢ ق

نوفمبر

two way binding [()

双向绑定

الـ two way إى اهل تغيره من الـ update من الـ ts ، ومن الـ ts الـ update يسمع من الـ html.

خطوات الـ two way data binding

١- لازم زعل module import من الـ app.module.ts ← root component ونعرفه من الـ 'import' property.
import {FormsModule} from '@angular/forms';
@NgModule({

imports: [
FormsModule
]
})

٢- بروح الـ input الـ value من الـ input ونحط [ngModel] ← banana in a box بين ngModel.ts property (& bind دسغل ngModel والـ)

<input [(ngModel)] = "userName" >
<p>{{userName}}</p>



اكتوبر

٢٠١٦/١٠/٣١

٢٠١٦/١٠/٣١

① directives

* اى حاجة تغيرها بـ ng يبقى اتابع الـ directive

* (١) اية دالة الـ **ngModel** من الـ **directive** تجعل **two way data binding**

(٢) يعني اية فعل **two way data binding** يعني بعده **input** يغير الـ **value** يتبعه **update** من الـ **ts** يعرض عن الـ **view**

(٣) اية دالة الـ **directive**

الـ **directive** في الـ **angular** هي حاجة يتغير
يا اما من تسلل الـ **html** / **html style** يتابع الـ **element**
يا لها في الـ **content**

٤ **content styling** مثلا ما زالت **directive**

٥ directives:

١- **ngClass , ngStyle** (to change styling)

٢- **ngIf**

٣- **ngFor** → structural directives

٤- **ngSwitch** لابد من بشرها

٥- **ngDo**

[logClass] = "{'classname': condition}" (٢)

"إذاً element ١٠ class لدينا ينطبق logClass وهذا

لذلك فهو يكتب كـ `logClass="classname > 10"` لأننا نريد أن يكون

العنصر الذي يزيد عن 10

أيضاً يمكن كتابة `logClass="classname < 10"` لـ `logClass="classname <= 10"`

ويمكن كتابة `logClass="classname != 10"` لـ `logClass="classname <> 10"`

أكتوبر

١٨ باب ٤٧٢٢ ق

١٤٣٧ هـ محرم ٥

② *ngFor

اد الـ **element** عاليـة من الـ **ngFor**

- "ngFor = "let x of y" → {{x}}
- "ngFor = "let x of obj" → {{x. -}}
- "ngFor = "let x of y; let i = index;"
- "ngFor = "let x of y; let isEven = even;"

لـ **index** لـ **true** بـ **even** وصـ

ex)

```
<tr *ngFor = "let u of users; let i = index; let e = even;>
```

```
<td [ngClass] = "'w-a': e"> {{i}} </td>
```

```
<td> {{ u.name }} </td>
```

```
<td> {{ u.age }} </td>
```

```
</td>
```

③ *ngIf

element لـ **true** بـ **boolean exp** ngIf لـ

فـ **element** لـ **false** دـ **else** سـ **hamboro** وـ **else** دـ **hamboro**.

- "ngIf = "condition"
- *ngIf = "condition; else hamboro"

يشـ كـ بـ هـ اـ الـ عـ الـ يـ يـ

لـ **false**, **condition** بـ

October

Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17



(4) [ngSwitch]

```
<div [ngSwitch] = "username">
  <p *ngSwitchCase = "'ahmed'"> -- </p>
  <p *ngSwitchCase = "'ali'"> -- </p>
  <p *ngSwitchDefault></p>
</div>
```

`<div *ngIf = "isThing ; then b1 ; else b2 "></div>`

`<ng-template #b1> — </ng-template>`
`<ng-template #b2> — </ng-template>`

`"ngFor = "let x of X ; index as i ; odd as o ; even as e"`

let i = index let o = odd let e = even

Day 15: ngFor

• ١٥ Friday, ٢٦ سبتمبر ٢٠١٥

• ١٥ Friday, ٢٦ سبتمبر ٢٠١٥

أكتوبر

١٦ باب ١٧٢٢ ق

١٤٣٧ هـ حرم ١٩

revision

8

ما هي الحاجة الى angular

9

لو ال backend ل project مختص بال

10

. frontend

apis من backend لو ال angular محتاج

11

FrontendBackend

12

Angular

APIs

1

ما هي نعمل الى backend لوحدة

2

backend او frontend

3

{ angular او react }

4

fullstack

ما هي architecture pattern

لو ال MVC او ال MVT

frontend

backend

no need for angular

MVC MVT

we need Angular

api

ما هي الميزة التي يجلبها angular

من backend

التفاعل سهل

October

Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24



١- الـ rest api : rest api الـ url .
 ٢- كل url /api / كل endpoint لـ url .
 ٣- كل url /api / كل endpoint لـ url .
 ٤- كل url /api / كل endpoint لـ url .
 ٥- كل url /api / كل endpoint لـ url .
 ٦- كل url /api / كل endpoint لـ url .
 ٧- كل url /api / كل endpoint لـ url .
 ٨- كل url /api / كل endpoint لـ url .
 ٩- كل url /api / كل endpoint لـ url .
 ١٠- كل url /api / كل endpoint لـ url .
 ١١- كل url /api / كل endpoint لـ url .
 ١٢- كل url /api / كل endpoint لـ url .
 ١٣- كل url /api / كل endpoint لـ url .
 ١٤- كل url /api / كل endpoint لـ url .
 ١٥- كل url /api / كل endpoint لـ url .
 ١٦- كل url /api / كل endpoint لـ url .
 ١٧- كل url /api / كل endpoint لـ url .
 ١٨- كل url /api / كل endpoint لـ url .
 ١٩- كل url /api / كل endpoint لـ url .
 ٢٠- كل url /api / كل endpoint لـ url .
 ٢١- كل url /api / كل endpoint لـ url .
 ٢٢- كل url /api / كل endpoint لـ url .
 ٢٣- كل url /api / كل endpoint لـ url .
 ٢٤- كل url /api / كل endpoint لـ url .
 ٢٥- كل url /api / كل endpoint لـ url .
 ٢٦- كل url /api / كل endpoint لـ url .
 ٢٧- كل url /api / كل endpoint لـ url .
 ٢٨- كل url /api / كل endpoint لـ url .
 ٢٩- كل url /api / كل endpoint لـ url .
 ٣٠- كل url /api / كل endpoint لـ url .
 ٣١- كل url /api / كل endpoint لـ url .

(١) الـ standard rest api ماشي مع الـ rest architecture style / rest API واحد و هو الـ standard .

- * rest api is api that follow REST architecture style .
- * rest is simplicity .
- * rest rules :
 - ① client - server model → use HTTP
 - ② stateless → interact by url
 - ③ uniform interface → urls
 - ④ layered system → client core
 - ⑤ cacheable → how client and server will communicate

الخميس الجمعة	السبت الأحد الاثنين الثلاثاء الأربعاء الخميس الجمعة	السبت الأحد الاثنين الثلاثاء الأربعاء الخميس الجمعة	السبت الأحد الاثنين الثلاثاء الأربعاء الخميس الجمعة	السبت الأحد الاثنين الثلاثاء الأربعاء الخميس الجمعة
١ ٢ ٣ ٤ ٥ ٦ ٧ ٨ ٩ ١٠ ١١ ١٢ ١٣ ١٤ ١٥ ١٦ ١٧ ١٨ ١٩ ٢٠ ٢١ ٢٢ ٢٣ ٢٤ ٢٥ ٢٦ ٢٧ ٢٨ ٢٩ ٢٩ ٣٠ ٣١				

October

ro

25

الأحد

Sunday

2015

أكتوبر

١٤٢٢ باب ١٤

١٤٣٧ هـ ١٥ صفر

component وـ components - think components

مقدمة الى components | web page

لـ ملف الـ html والـ css والـ js (اصنـ بيـ دـ)

scalability وـ reusability | مـ دـ

class للـ behavior وـ functionality

class decorator

11

component | class يـ مـ خـ اـ

built-in decorators | runtime

@Component({})

1

→ angular interpolation:

→ interpolation refers to embedding expression
into markup text.

³ (ex) <h1>{{ }}</h1>

→ <h1>{{ user.username }}</h1>

<h1> {{user?.username}}</h1>

5

→ ? is (safe navigation operator)

used to check if the object
is null or not.

→ if user object is not null

then get the username.

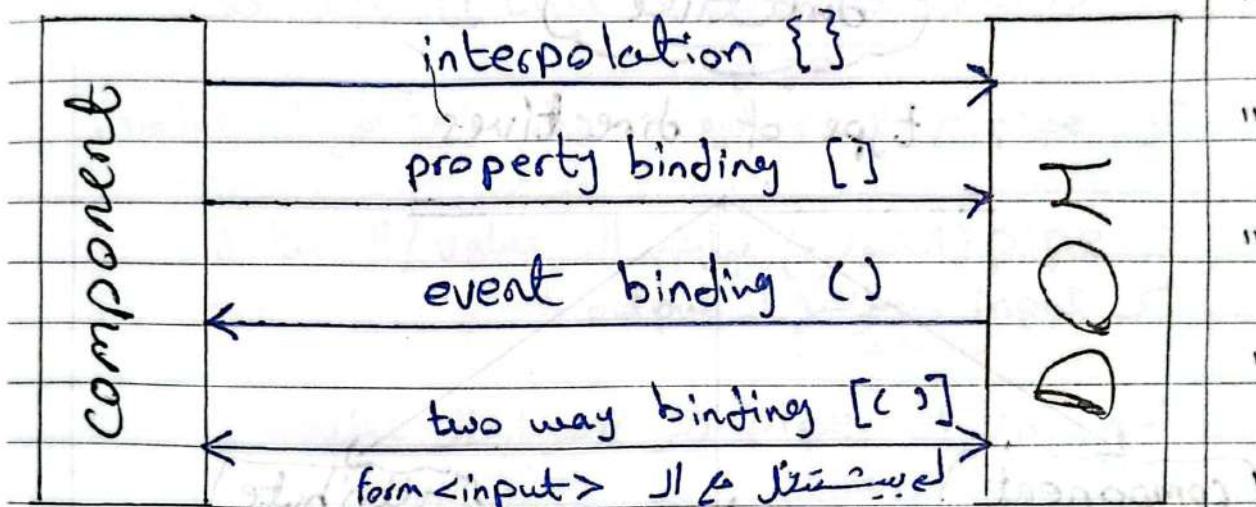
else don't get anything.

October

Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

Data binding ٩٧

الـ data binding هو إن أخذنا نعرض معلومات الـ component من ملف الـ ts إلى الماء بـ HTML.



لـ `value` الـ `one way` binding من الـ component يحصل على الـ `value` من الـ `html / dom` ويسعى من الـ `property binding` والـ `interpolation`.

او التغيير يحصل من الـ `html` ويسمى `two way binding` من الـ component event binding (ex)

من الـ `dom` يحصل من الـ `component` `two way binding` والـ `العكس`.

الخميس الجمعة	السبت الأحد الاثنين الثلاثاء الأربعاء الخميس الجمعة	السبت الأحد الاثنين الثلاثاء الأربعاء الخميس الجمعة	السبت الأحد الاثنين الثلاثاء الأربعاء الخميس الجمعة	السبت الأحد الاثنين الثلاثاء الأربعاء الخميس الجمعة
٣١ ٣٠ ٢٩ ٢٨ ٢٧ ٢٦ ٢٥ ٢٤ ٢٣ ٢٢ ٢١ ٢٠ ١٩ ١٨ ١٧ ١٦ ١٥ ١٤ ١٣ ١٢ ١١ ١٠ ٩ ٨ ٧ ٦ ٥ ٤ ٣ ٢				

أكتوبر

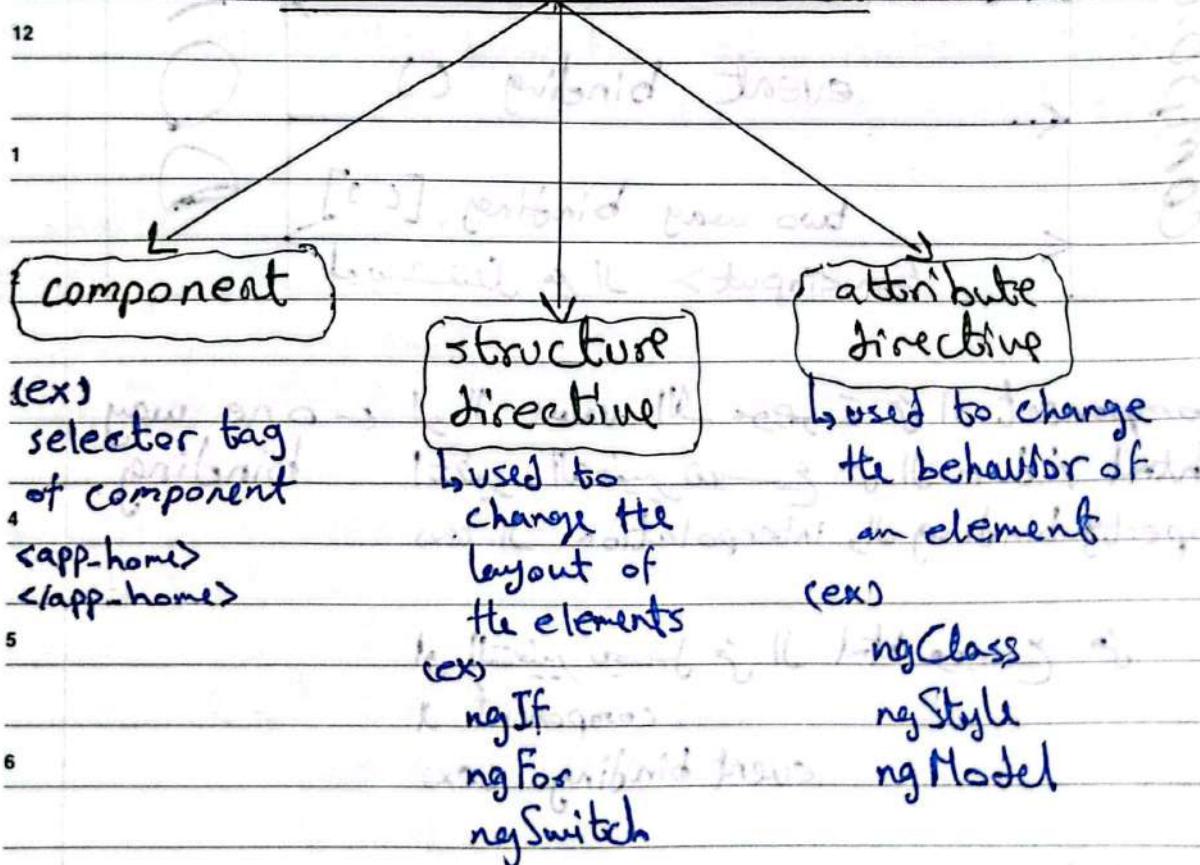
١٢ باب ١٧٢٢ ق

١٤٣٧ هـ محرم ١.

٧. tags أو attribute و directive لا يرجع الى html لكن يستعمل من اد
٨. element style / design بما ينبع من الـ
٩. content او الـ أضافة من إضافة
١٠. component بما ينبع من content او أضافة

directive

type of directives



October

Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat							
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31



ال pipe حقيقة لا directive لكن الامر :
 « ال directive ينفذه على element معين .
 ~ value ~ ~ pipe »

ال built-in pipes يتوفر في angular

(ex) `<p> {{ title | uppercase }} </p>`

كل value من ال pipe يدخل حاضر في ال angular

ال data / value pipe يأخذ ال data ويجعل output

ال الزمن من ال pipe هو تحويل ال

`value → (pipe) → transformed value`

• ~~the value can be converted to another type
using the pipe operator. It's like a transformation function.
It transforms the input value into the output value.~~

• ~~Transformation is also called piping in Java
which is the second thing I learned.
In Java, we have two types of transformation:
1. Direct transformation
2. Indirect transformation~~

الخميس الجمعة	السبت الأحد الاثنين الثلاثاء الأربعاء الخميس الجمعة	السبت الأحد الاثنين الثلاثاء الأربعاء الخميس الجمعة	السبت الأحد الاثنين الثلاثاء الأربعاء الخميس الجمعة
٢٢ ٢٣ ٢٤ ٢٥ ٢٦ ٢٧ ٢٨ ٢٩ ٣٠ ٣١	١٠ ١١ ١٢ ١٣ ١٤ ١٥ ١٦ ١٧ ١٨ ١٩ ٢٠ ٢١ ٢٢ ٢٣ ٢٤ ٢٥ ٢٦ ٢٧ ٢٨ ٢٩ ٣٠	١٠ ١١ ١٢ ١٣ ١٤ ١٥ ١٦ ١٧ ١٨ ١٩ ٢٠ ٢١ ٢٢ ٢٣ ٢٤ ٢٥ ٢٦ ٢٧ ٢٨ ٢٩ ٣٠	١٠ ١١ ١٢ ١٣ ١٤ ١٥ ١٦ ١٧ ١٨ ١٩ ٢٠ ٢١ ٢٢ ٢٣ ٢٤ ٢٥ ٢٦ ٢٧ ٢٨ ٢٩ ٣٠

October

③ Angular routing

8 www.example.com / home / ...
 domain name path

9 /home display HomeComponent
 10 /about display AboutComponent

11 If the path is 'home', display the 'HomeComponent'.
 In the place of <router-outlet></router-outlet>

12 ما هي فكرة الـ routing ؟
 1 (navigate) is

2 عُملية الـ routing بين المُكونات

2 (browser url) معنٍ مُباشرة على url path

3 مثلاً (route) /url path في component

4 لهمار، طريق و بنو صدر من خلاة.

5 component directive هي عبارة عن <router-outlet>

6 وظيفتها إنشاء بستان و يربط مكانة الـ selector

component كل معنٍ path على component

6 منطقة path خاص بها.

7 built-in attribute معه routerLink = "url".

8 يترك عن الـ href = "url" attribute

9 browser URL path لـ link

10 page reloading من تغير ما يحصل.

١٩

19

الاثنين

Monday

2015

أكتوبر

باب ١٧٢٢ في

١٤٣٧ هـ

١- {path: '', component: []}

٢- routerLink = ""

٣- <router-outlet></router-outlet>

٤- `component` لـ `redirect` فاصل زغل path
محسن كـ

١٠

فترة طريقة :

٥- `component` زغل path متر خاص لنفس الـ `path` (١)

١٢

وذلك من الـ best practice لـ `best` امـ `load`
لنفس الـ component .

٦

{path: "", component: HomeComponent},

٢

{path: "home", component: HomeComponent}

٣

٤- `path` لـ `redirect` نفس `path` (٢)

٤

{path: "", redirectTo: "home", pathMatch: 'full'}

{path: "home", component: HomeComponent},

٥

احـ هوـ الـ {pathMatch}

٦- `redirectTo` تـ `path` لـ `path`
ـ `url patterns` بـ `url`

path: "home",

path: "home/setting"

path: "home/category"

October

Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

① in navbar.component.html

```
<a [routerLink] = "home" > Home </a>
```

② in app-routing.module.ts interface

```
const routes: Routes = [
```

```
{ path: 'home', component: HomeComponent }, ]
```

③ in app.component.html

`<app-navbar></app-navbar>` url
`<router-outlet></router-outlet>` selector
`<app-footer></app-footer>` component
 ينتمي مسار url
 .path

localhost:4200 / home

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

instruction ٥ browser URL يتعامل مع ال routing
 instruction view بناءً على ال view

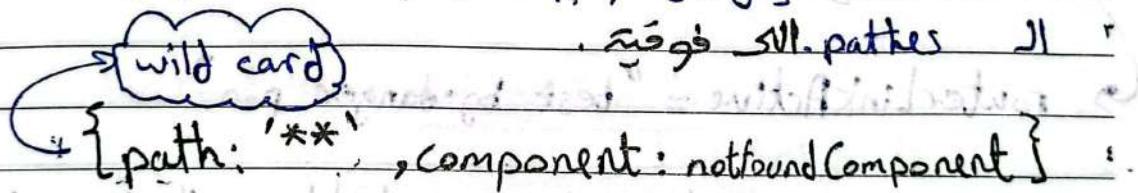
redirect ← يعنى الـ path الـ `pathMatch: 'full'`
 لأنّه يكون نفس الـ text مكتوب في
 الترتيب هنا
 الـ `"redirectTo": "https://www.google.com"`

redirect ← يعنى اعمل مع أول path بـ `pathMatch: 'prefix'`
 أول كلمة فيه نفس الـ text مكتوب في
 الـ `"redirectTo": "/user/**"` تبيّن
 محيط الـ path

إذاً لو ادخل user كـ path، نُظّم له صفحة
 محيطة بـ `"/user/**"` عن طريق الـ wild card

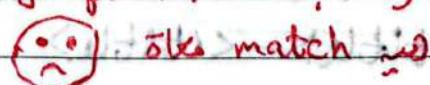
(١) أولاًً أعمل على `create` عادي جداً
 - (`notFound`) سمعت مثل:

(٢) نعمل للـ component دالة `path: '**'` وردة
 معناها يعنى أي حاجة مني جست جات باقى
 الـ path فوقيه.



`{ path: '**', Component: notfoundComponent }`

(٣) لازم لازم الـ `path: '**'` يكتب آخر path وردة
 لأنّ الـ route يُتم على `path` من خوق لحتي
 بالترتيب واحد واحد، الـ route يُتم على أول path
 يُتحقق هلي بـ `match` الـ url path ولا لا، لو لاد
 فـ `notfound` مالك بعده وهذا ... لغائية ما يحصل للـ `(path-all)`
 لو كتبنا `path: '**'` الاول، كده اى url يُتحقق

 match

IV

17

السبت

Saturday

2015

أكتوبر

٦٦٢٢٠١٧

٤٤٣٧ هـ

٩. active نص معين للـ link style \rightarrow active router لـ

① in home.component.html

١٠ < a routerLink = "home" routerLinkActive = "test" > click < /a >
 ↓ ↓
 ١١ path ارت class ارت

② in home.component.css

12

.best {

1

}

2

class يربط خصائص routerLinkActive لـ

routerLinkActive = "test bg-danger p-0"

3

أعلى قيم لـ < head > < title > لـ component لـ

4

{ path: 'home', component: HomeComponent, title: 'home' } وـ

title طريقه نقدر نغير ال property له من طرفي او

component لـ

< title > - < /title >

October

Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

nested routes

path ↳ path يعني نقل nested route ٰ
component ↳ components

(children) ↳ property

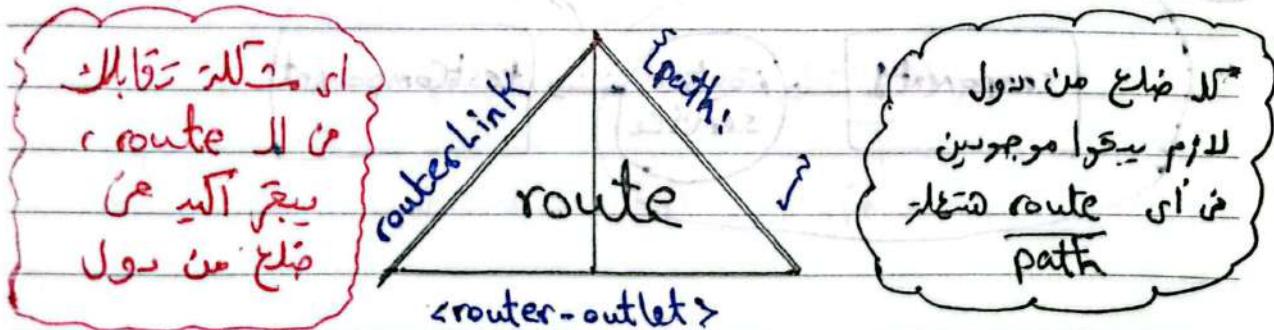
```
(ex) {path: 'settings', component: SettingComponent,
      children: [
        {path: 'profile', component: ProfileComponent},
      ]},
```

داخل الـ **SettingComponent** path ↳ routerLink وينفذ الـ browser URL

localhost:4200/settings/profile

↳ nested route in path → (settings)

ـ راجعاً وابداً الـ route هو مسار من متعدد ملائم .



10

15

الخميس

Thursday

Router
service{ set parameter
to url }

2015

أكتوبر

٤ باب ٢٢ ق

١٤٣٧ هـ سبتمبر

Route parameter Router service

① {path: 'products/:id', component: testComponent}

② inject Router service.

router | navigate is library service |
. . component | /typescript | ثان عن داخل ال

```
constructor(private router: Router) {}
```

③ gotoTestcomponent(product)

```
let x = product.id;  
this.router.navigate(['/products', x]);  
↳ /products/x
```

④ <div (click)= "gotoTestcomponent(x)"

"ngFor = "lets x of products" >

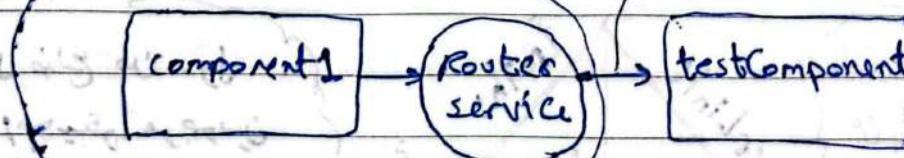
<h1> {{x.id}}, {{x.name}} </h1>

</div>

5

/products/x

6



October

Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

لِزَانِي تَحْرَارُ الـ route الـ parameter الـ جَاهِي مَعَ الـ URL حَتَّى الـ parameter الـ جَاهِي مَعَ الـ route

عن طَرِيقِ ActivatedRoute . service اسْتَعْمَلْ

① constructor(private ar:ActivatedRoute){}

~~② ngOnInit(): void {~~ بِسَبِيلِ string لِـ route

~~let x = this.ar.snapshot.paramMap.get('id');~~ " جَاهِي الـ route

~~let y = parseInt(x);~~ جَاهِي الـ integer

~~this.prop = parseInt(this.ar.snapshot.paramMap.get('id'));~~

الـ snapshot مُكَلَّمةٌ إِنَّهَا يَسْتَدِعُ مِنْ وَجْهِي وَعَنْ
بِحِسْبِ بَاعِي تَغْيِيرِ جَاهِي عَارِي

بِالْتَّالِي أَهْنَا لِلزَّمْرِ تَرَاقِبَ الـ parameter عَارِي بِعِنْدِ .subscribe

~~③ ngOnInit(): void {~~ نَتَحْتَمُ الـ observable وَنَعْلَمُ

this.ar.paramMap.subscribe(
(x: paramMap) => {
this.prop = parseInt(x.get('id'));
})

أَوْ تَغْيِيرِ ادِيلَكِ notification

Session 4

٩- Components life cycle

١٠ components interaction

١١ angular services

١٢ dependency injection

١

٢

٣

٤

٥

٦

① component life cycle

* الـ **component** سيعمل لـ **life cycle** يتم إدارته بواسطـة **angular** .
نهاية دورة **life cycle** من أول ما يفتح لغاية ما يغلـ .
دورة تنتهي حالـ **Component** من أول ما يفتحـ .
نهاية دورة **Component** من أول ما يـ **access** .
* **Component** هل يمكن من الأول ما يـ **access** ؟

* كل **Component** سيعمل لـ **lifecycle** / لـ **مراحل** بعد بدء :
ـ بـ **angular** إدارة المراحل وـ **مهـ** من طرـيـ الـ **angular** :

- ① angular create the component.
- ② angular renders it.
- ③ angular create & render its child component.
- ④ angular check the component when its data-bound properties change.
- ⑤ angular destroy the component before removing it from the DOM

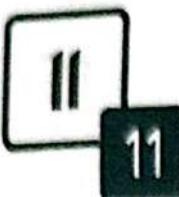
* **lifecycle hooks** يـ **توفرـ** عن **angular** **لـ** **run** **على طـيـقـهـ** **من قـرـنـ** **فـتـلـ** **مـجـمـعـ** **lifecycle** **عـنـصـرـ** **عـنـصـرـ** **عـنـصـرـ** **عـنـصـرـ** .

↳ **Angular 7** **hooks** **new** **in** **action** ↳ **hooks**

↳ **transient** **with** **parent** ↳ **transient**
↳ **transient** **with** **parent** ↳ **transient**
↳ **transient** **with** **parent**

الخميس الجمعة	السبت الأحد الاثنين الثلاثاء الأربعاء الخميس الجمعة				
٢١	٢٣ ٢٤ ٢٥ ٢٦ ٢٧ ٢٨ ٢٩ ٢٩	٢٦ ٢٧ ٢٨ ٢٩ ٢٩	٢٨ ٢٩ ٢٩ ٢٧ ٢٦	٢٩ ٢٨ ٢٧ ٢٦ ٢٥ ٢٤	٢٥ ٢٤ ٢٣ ٢٢ ٢١ ٢٠ ١٩ ١٨ ١٧ ١٦ ١٥ ١٤ ١٣ ١٢ ١١ ١٠ ٩ ٨ ٧ ٦ ٥ ٤ ٣ ٢

October



الأحد

11

Sunday

2015

أكتوبر

٢٠١٧٢٢ نو吐

١٤٣٦ ذوالحججة ٢٨

٧ اول حاصل يستخرج من ال component دوال دل بداع
component اوال class بداع ال component

٨ اية دو ترتيب ترتبيه ال methods في ال component class

٩ اول دو ترتيب بيتخرج من ال component method
بعد كل functions ينبع من component دوال
وكل function ينبع منه ترتيب دوال مترافقه من
مراحيل دو رتيبة ال component

١٠ اول دو ترتيب دوال lifecycle hook بعد
بعد كل call ويتبعه constructor
component دوال input parameter
(@Input decorators) دوال property دوال
(@Input decorator) دوال input data دوال

١١ اول دوال ينبع من call دوال ngOnChanges()
بعد دوال call ويتبعه ngOnInit() دوال
اند دوال initialized ينبع component دوال

١٢ اول دوال ينبع من call دوال ngDoCheck()
call دوال ngOnInit() دوال ngOnChanges() دوال
عن اى change detection دوال

١٣ اول دوال ينبع من call دوال ngOnDestroy()
use دوال destroy دوال component دوال
من component ل component دوال
عن دوال component دوال

October

Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat							
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Q) أين دور الفرق بين الـ `(constructor)` والـ `(ngOnInit)` ؟
 طالما الاثنين يمتلكان `call` الى الـ `component` يفتح
 ؟ initialized component // اولاً

الـ `(constructor)` دائمًا ي被执行ة أول وفقط
 الـ `(ngOnChanges)` ي被执行ة call بعد الـ `(ngOnInit)`
 مما يجعله.

نستخدم اسفل لابد دعوه الى `best practice`
 " الـ `constructor` لا تخدم من حاجته فقط ،
 ① من `variable initialization`
 ② في الـ `dependency injection`
 ان حاصي غير الاثنين دول عالياته يستخدموها اولاً الـ `ngOnInit`
 يفتح بعده يظهر في الـ `(constructor)`

Q) ايه هي الهمية الـ `lifecycle hooks` ؟

أهمية ان احنا نقدر نعمل run لكور محبين على مرضلهم
 معيته من مراحل تستغل الـ `component`

(مثال) فخذ الكودة طالما `component` يفتح.
 وله مدة معرفة وهو يدخل.

• `@Input property` لا يحصل تغيير عن قيمة

ان شاء الله كل فتحة الـ `api` هو الـ `(ngOnInit)`.

الخميس الجمعة	السبت الأحد الاثنين الثالثاء الاربعاء الخميس الجمعة	السبت الأحد الاثنين الثالثاء الاربعاء الخميس الجمعة	السبت الأحد الاثنين الثالثاء الاربعاء الخميس الجمعة
٢١	٢٢ ٢٣ ٢٤ ٢٥ ٢٦ ٢٧ ٢٨ ٢٩ ٢٩ ٣٠ ٣١	١٦ ١٧ ١٨ ١٩ ٢٠ ٢١ ٢٢ ٢٣ ٢٤ ٢٥ ٢٦ ٢٧ ٢٨ ٢٩ ٣٠ ٣١	١٦ ١٧ ١٨ ١٩ ٢٠ ٢١ ٢٢ ٢٣ ٢٤ ٢٥ ٢٦ ٢٧ ٢٨ ٢٩ ٣٠ ٣١

October

7 ابراهيم نسخة الـ lifecycle hooks

8 interface لـ lifecycle hooks. كل function من الـ interface خاص بي، يعني كل function موجود في ..

9 مثلاً.

بالتأكيد عتاه نقدر نستخدم أي function

10 للزوم نعمل implement للـ interface بـ `implements` عن طريق الـ component class.

11

12 لـ `ngOnInit` او `fire` `ngOnChanges()`

لأنه دخل الـ sequence والطريق المطلوب والطريق الآخر مرتاح، أنا لازم أعرف أية التغيير الذي حصل من الـ data بـ `SimpleChanges` قبل ما نفرون الـ component.

13 اية دو لا زمة الـ SimpleChanges

14 `ngOnChanges(changes)` دخل `ngOnChanges()`

15 `changes` من النوع `SimpleChanges` معروفة `SimpleChanges` changes

16 `input data` عن `object` يحتوى على الـ `changes` قبل دخل `SimpleChanges`.

17

```
18 . ngOnChanges(changes: SimpleChanges) {
    console.log(changes);
}
```

19 }

② Components Interaction

sharing هو أن أتمكن من مشاركة components interaction

البيانات بين المكونات

عكن نعم components interaction

① nested component interaction

child component هو المكون الممتد داخل المكون

nested component داخلي المكون

لو المكون الممتد يأخذ بيانات / يفرجها

@Input من المكون الممتد

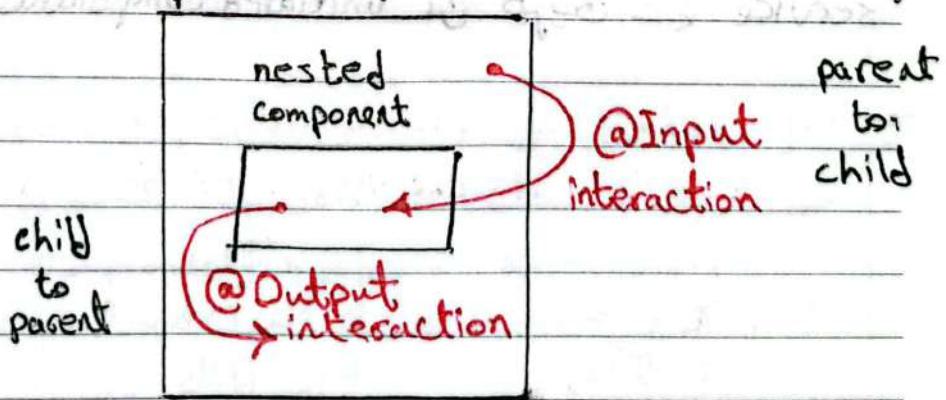
لو المكون الممتد يفرجها

@Output إلى المكون الممتد

والمكون الممتد يفرجها

nested component بين المكونات

parent component



V

7

الأربعاء

Wednesday

2015

أكتوبر

٢٦٢٢ ق

٢٤ ذو الحجة ١٤٣٦ هـ

7

unrelated components
interaction

②

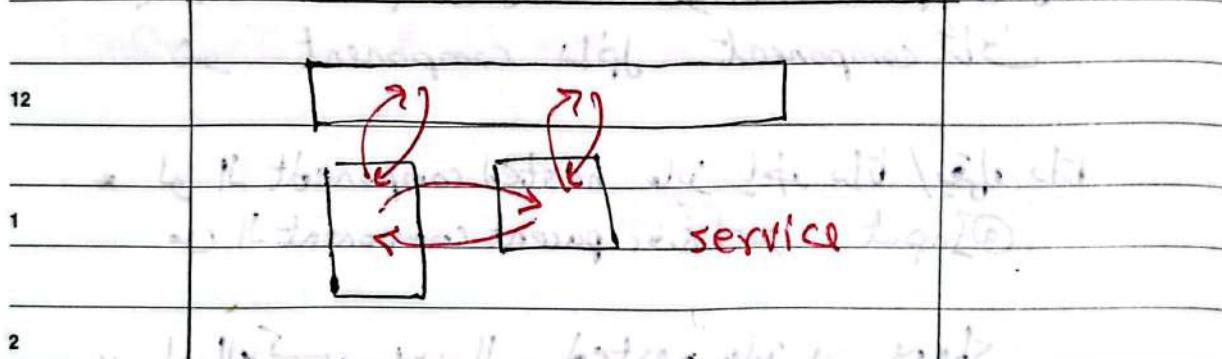
8

وده عبارة عن ان احنا نعمل sharing للبيانات
 بين components مارغون ملاقة بعضهم
 . nested . يمثل متى

10

ما بنعمل service لل sharing

11



3

ما بنعمل data لل sharing بين
 @Input() ← من طريق nested components ←
 @Output()
 service ← من طريق unrelated components ←

ملاحظة

October

Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

١ share data from parent to child :

at parent.component.ts

```
export class ParentComponent {
  selectedID: number = 0;
```

at parent.component.html

```
<select [ngModel] = "selectedID">
```

=

</select>

٢ <app-child [idfromparent] = "selectedID"></app-child>

at child.component.ts

```
export class ChildComponent {
```

① @Input() idfromparent: number = 0;

}

• أولاً بـ @Input() property decorator قبل .parent من الـ data الذي يتقبل من child property

٣ اعمل binding بين ملف الـ ts بين parent والـ child من طريق الـ selector المضاف بالـ child component

```
<app-child [idfromparent] = "@Input()">
```

اسم الـ property = الـ idfromparent

الـ child للـ child

٥

5

الاثنين

Monday

2015

أكتوبر

٢٤ سبتمبر ٢٠١٥

٢٢ ذي الحجة ١٤٣٦ هـ

child.ts

<app-child [child] = "parent" >

parent.ts

</app-child>

الخواص (Properties) هي امثلة على اسفلها من الـ API

child ، او للاختصار child مكان من الـ API

وارجعه @Input decorators

بعد حفظ / ثم وانما نهاد child جوا الى

- يكتب اسم الـ property الذي يحتوي على child

- وأنتاب الماء الذي يحيط بالـ child

للحاجة نقل child من parent الى child من child

ينتقل child من parent الى child

عن طريق request

parent يرسل child لـ request

request يطلب child من child

child يعطي request child

بالنهاية request يحصل على child

تنتقل data من child الى parent

عن طريق request

- ① احجز مكان للستقبال data عن طريق @Input
- ② ابحث data من مكانه في parent ملائمه child
- ③ من خلال component selector بحث child

October

Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat							
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

① `@Input() child: any = 1;`
 ② `<app-child [child] = "P" ></app-child>`

(ما هو ؟) أي تغيير يحصل على `@Input()` ، فسيحصل `ngOnChanges()` call

الـ `value` سيغير ترتيبه إن الـ `ngOnChanges()` اتى

data للـ sharing. هناك 3 cases

- ① أعمل share data من الابن لابن
- ② أعمل share data من الابن لابن
- ③ أعمل file service كل الابن

مفيه share بين siblings
 هو share يا إما بين الابن لابن ،
 يا إما بين الابن لابن ،
 يا إما file بـ share لكلثة.

٣

السبت

3

Saturday

2015

أكتوبر

٢٢ سبتمبر ١٤٣٦ھ

٢٠ ذوالحجjah ١٤٣٦ھ

- ٧ ② share/send data from child component to parent component.

٨ الـ child component من الـ data منتقل
 ٩ data الـ child من الـ parent component
 . child component من الـ موجود

١٠ at child.component.ts

١١ export class ChildComponent { } .

١٢ ① @Output() childEvent = new EventEmitter();

١٣ parent الـ child من الـ childEvent
 . @Output() childEvent = new EventEmitter();

١

٢ بعد كذا يأتي اسم طارىء وعده Function .

٣ الـ childEvent من ملف الـ html للـ function

٤ الـ button من html من event

٥ ومن المعلومات التي تجدها من الـ parent الـ child

٦ ③ event(child) { }

٧ this.childEvent.emit("the data");

٨ }

٩ هنا نعطي المعلومات الى

١٠ parent نعبر الـ

١١ }

October

Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat							
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

٣- هنا نحن الـ function الـ `child` نتبع البيانات من
أو child أو parent في ملف الـ html الخاص بلا child
at child.component.html

③ `<button (click) = "eventChild()">`
go to parent
`</button>`

لـ الـ user يضغط على الزر ، الـ function يـ `call` الـ function
data / إيمانـ `emit` جـ `function()` حـ `parent`

٤- نزوح الـ parent ونـ `parent` فـ `child`
المعلومـ `child` جاءـ من الـ

export ParentComponent {
④ message: string = "";
}

٥- نزوح طبقـ الـ parent اـ `parent` ونـ `parent`
، child component .
من خـ الـ child component selector `<app-child>`
الـ property @output property binding
data فـ `parent` فـ `child`

⑤ `<app-child (childEvent) = "message = $event">`
`<app-child>`
`<p>{{message}}</p>`

أكتوبر

٢٠١٧٢٢ نوٰت

١٨ ذو الحجه ١٤٣٦ھ

③ angular service

- ٨ angular JS مفهوم service اد عبارة عن دالة يحتوى على data (مقدار معين) بين أكثر من component (ملحوظة) يُمكن share بينها

٩

١٠ اية ايه ال service ؟ اية هو ال service

١١

business logic || file Ts هو service وال data المترتبة تكون shared بين components

١٢

اذا نعمل بال CLI لـ service

١٣

» ng g s userService

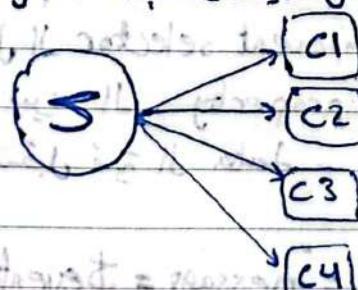
١٤

اهى ندخل الـ service

١٥

لـ service يمكن اخذ functions او data من خارجها من reusability أكثر من component يمكن تحقيقها

١٦



١٧

لـ service يمكن اخذ data

١٨

او function

١٩

من أكثر من component

٢٠

و properties في class service اد component يمكن اخذها من أي methods

October

Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

ما هو الـ service decorators
ما هو الـ @Injectable decorators
ما هو خاصيته؟

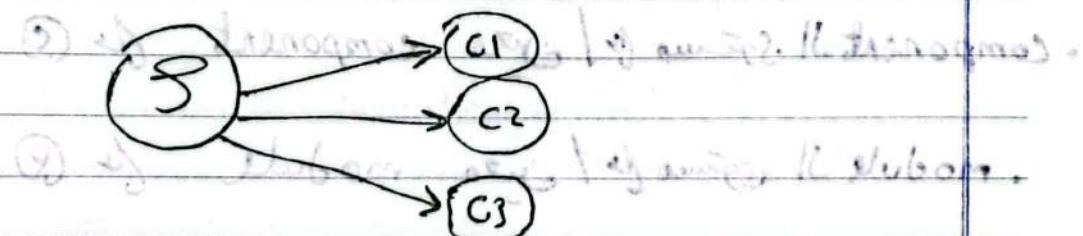
الـ @Injectable decorator يسرّف الـ class كـ service في الـ angular ويسمح له inject في الـ component من اى service بالثانية access الـ component

ما هي الـ 3 ways الـ service will provide

أولاً: ان احنا نعمل على service يعني نحدد scope من يقدر عليه الـ access الـ service .

معننا 3 أماكن / 3 decorators يعني نحدد service will providing .

① بداخل الـ @Injectable عن طريق providedIn: 'root' ورها يمكن الـ accessible service ويتبع نسخة واحدة متراكمة ما بين الـ components يعني اي تغيير في الـ service يتبع كل الـ components

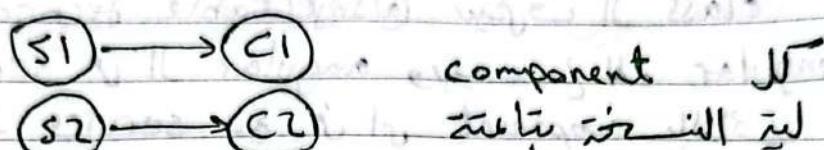


سبتمبر

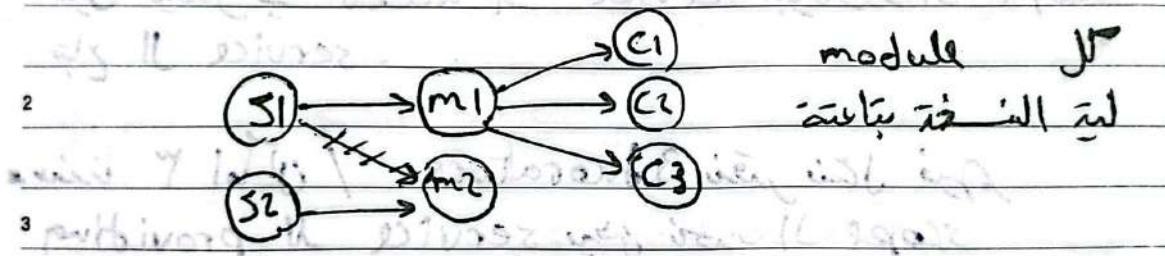
١٨ ثوت ١٤٣٢ في

١٦ ذوالحججة ١٤٣٢هـ

- ٧ بداخل الـ `property` @Component عن طريق `providers: [service]`
 ٨ اسم الـ component ، وردة دخواں الـ providers: [service]
 ٩ لئے السخن الخاتمة بیٹھ من الـ service، یعنی ای تنہی
 ١٠ ایم الـ components میں دیکھ سکتے ہیں باہم الـ service
 ١١



- ١ بداخل الـ `property` @NgModule عن طریق `providers: [service]`.
 ٢ ایم الـ module ایم الـ service، وردة دخواں الـ providers: [service].
 ٣ مسحولک الـ module کو تبع الـ components نہ دکھلے.



- ٤ service نہ 3 scope line پر مسحولک
 ٥ project نہ / root ایم الـ

• component میں / مسحولک component

• module میں / مسحولک module

September

Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Sat	Sun	Mon	Tue	Wed
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21



(ex)

① >> ng g s users ↵

② at users.service.ts

@Injectable({ providedIn: 'root' })

export class UsersService {

users: object[] = [{ id: 1, name: 'Ahmed', age: 30 }];

③ at home.component.ts

export class HomeComponent {

homeUsers: object[] = [];

constructor() {

let x = new UsersService();

this.homeUsers = x.users;

اتبعوا راهناً اي service اوي تستعملها ، اغلى من

نسخة بادل يعني بيغيز اغلى نسخة من

غير ما تطلع على طريقي استخدام

Dependency injection .

constructor(private _x: S) {}

constructor(private x: UsersService) {}

this.homeUsers = this.x.users;

سبتمبر

١٦ أكتوبر ٢٠١٥

١٤ ذوالحججة ١٤٣٦ هـ

④ dependency injection

ما هو الـ design patterns

أولاً، يعنـى pattern يعني طريقة أو اسلوب design.

ثانياً، يعنـى اسـاطـة تـقـمـيم أو طـرـيقـة مـخـتـلـفـة لـالـتـعـيمـ

ومـقـصـورـ بـالـسـعـيمـ هوـ تـقـمـيمـ الـأـنـظـارـ

ثالثاً، classes project إلى متـفـقـاتـ الـf~ld~esـ إـذـاـيـ الـc~lass~sـ وـالـc~omponent~sـ

12

أـنـوـاـنـدـ design patterns

1

ـ حلـولـ مـعـيـنةـ خـارـجـ

ـ الـd~e~v~e~l~o~p~e~r~sـ الـd~e~v~e~l~o~p~e~r~sـ الـd~e~v~e~l~o~p~e~r~sـ

ـ الـd~e~v~e~l~o~p~e~r~sـ الـd~e~v~e~l~o~p~e~r~sـ الـd~e~v~e~l~o~p~e~r~sـ

ـ الـd~e~v~e~l~o~p~e~r~sـ الـd~e~v~e~l~o~p~e~r~sـ الـd~e~v~e~l~o~p~e~r~sـ

ـ كلـ الـd~e~v~e~l~o~p~e~r~sـ الـd~e~v~e~l~o~p~e~r~sـ الـd~e~v~e~l~o~p~e~r~sـ

ـ منـ حـمـنـ الـd~e~v~e~l~o~p~e~r~sـ الـd~e~v~e~l~o~p~e~r~sـ الـd~e~v~e~l~o~p~e~r~sـ

ـ يـخـرـمـاـنـ الـd~e~v~e~l~o~p~e~r~sـ الـd~e~v~e~l~o~p~e~r~sـ الـd~e~v~e~l~o~p~e~r~sـ

September

Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

هائل من خطة من الـ service
من غير ما تعرّب مالـ constructor

dependency injection

٢٠١٥

السبت

Saturday

٢٦

١٣ ذوالحجّة ١٤٣٦ هـ

١٨٩٢ ق ٢٢٠١٧

سبتمبر

ما هو الـ dependency injection pattern

روي عباري عن design pattern بجملة الاعمارية بين الـ classes ، عن طريق انه يدخل الـ class مستقل عن الـ classes الا بسيطة ملحوظ .

هو programming technique بين الـ objects والـ classes .

لربّن تخرّج عن الـ Angular .

الـ service هي أكثر من service عن طريق ان احنا نستعمل كل الـ component ، بالذات كل الـ component من الـ constructor مسخة على الـ service ، يعني ان اوي تغير من الـ constructor يغيّر مثلاً كل الـ components التي يتعلّق عليه .

اكل ان احنا ن Inject او inject الى الـ component من غير ما نطلع على الـ constructor لـ calling .

الـ constructor يتبع الـ component من غير ما نطلع على الـ service من غير ما نطلع على الـ service .

(ex constructor (obj : UserService) {

this.comps = obj.users ;

}

٣٢	٣١	٣٠	٢٩	٢٨	٢٧	٢٦	٢٥	٢٤	٢٣	٢٢	٢١	٢٠	١٩	١٨	١٧	١٦	١٥	١٤	١٣	١٢	١١	١٠	٩	٨	٧	٦	٥	٤	٣
٣٠	٢٩	٢٨	٢٧	٢٦	٢٥	٢٤	٢٣	٢٢	٢١	٢٠	١٩	١٨	١٧	١٦	١٥	١٤	١٣	١٢	١١	١٠	٩	٨	٧	٦	٥	٤	٣	٢	

September

٥٠

٢٥

الجامعة

Friday

2015

سبتمبر

١٤٧٢٢ ق

١٢ ذوالحججة ١٤٣٦ هـ

(ex)

export class HomeComponent {

let X:object[] = [];

① constructor () {

let _UserService = new UserService();

this.X = _UserService.users;

11

② using dependency injection

يمثل هذا إعلان حقن الخواص - الكلا

محظوظ على أي من برة برة ، يعني من غير ممكن

للـ constructor بـ كل خواص

constructor (_UserService:UserService) {

this.X = _UserService.users

{}

implementation في angular للـ ما هو ظهر

dependency injection للـ

design pattern للـ

خارج الـ angular



الـ dependency injection هو أسلوب مخلص ومربي

لـ دوامات خدعة وفلترة تافهة

قوى عليه

September

Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed
١	٢	٣	٤	٥	٦	٧	٨	٩	١٠	١١	١٢	١٣	١٤	١٥	١٦

١١١١٢٠٦٤

الذئب

Thursday

٢٤

٢٠١٥

١١ ذوالحججة ١٤٣٦ هـ

١٢ ثوبت ١٧٢٢ في

سبتمبر

• يفضل نقل الخدمة من ال طريقي
ال dependency injection

• يفضل الخدمة بيئة ال service تكون
لرئاستها من هن تخديها خارج ال class

constructor(private s1:S1,
private s2:S2){}

"ما هو مفهوم constructor؟"

constructor parameter أو يسيطر على access modifier
وهي مفتاح إن ال parameter هي variable متغير خارج
constructor parameter وتحتاج assign بقيمة ال constructor داخل ال constructor
، TS أو behind the scene وكل دة

(ex)

public x:string;
constructor(x:string){
this.x = x;
}

↑ constructor(public x:string){}

طبع دة (٢٠٢٠) فتحت
أن أنا أعرف ال x واتفتح
باليدي.

angular بناء تخدم أو service
أعمل منه نسخة بال dependency injection
، class و component ، service

٢٣

الأربعاء

23

Wednesday

2015

سبتمبر

١٢ سبتمبر ١٤٣٦ هـ

١ ذوالحججة ١٤٣٦ هـ

٧ APIs

Session 5

٨ ١- Observables.

٩ ٢- HttpClient.

١٠ ٣- template reference variable.

٤- pipes

٣

٤

٥

٦

٧

٨

September

Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed
١	٢	٣	٤	٥	٦	٧	٨	٩	١٠	١١	١٢	١٣	١٤	١٥	١٦	١٧	١٨	١٩	٢٠	٢١	٢٢	٢٣



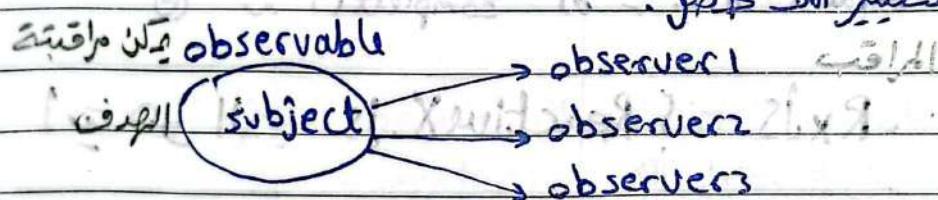
① Observables

ما هو ال observable

ال observable هو بديل لل promise بين خدمة
لأن ن RJF ال asynchronous code يعني ن RJF
ال sync code ل asynchronous code

وال observer design pattern يتبع observable

وهو ياخذ مثمار عبارة عن علاقة one to many بين ال objects
يعنى فيه object ومتى يعزم عليه change objects ، بالتالي
ال object إذا حملة تتغير كل ال objects الذي معهمه على
نفسها بالعكس الك خطأ



يعنى باختصار ال object تغيرات ال object الذي معهمه

امثلة على ال asynchronous code
api call
ajax call
fetch call

(اللامسة) ال observable من خدمة طرافية أو

الاک يستخرج من httpclient asynchronous code

لأن نستخدم ال observable لازم نسترك

فتة بعض رسالة subscribe

الثلاثاء الأربعاء الخميس الجمعة	السبت الأحد الاثنين الثلاثاء الأربعاء الخميس الجمعة	السبت الأحد الاثنين الثلاثاء الأربعاء الخميس الجمعة	السبت الأحد الاثنين الثلاثاء الأربعاء الخميس الجمعة
٣٠ ٢٩ ٢٨ ٢٧ ٢٦ ٢٥ ٢٤ ٢٣ ٢٢ ٢١ ٢٠ ١٩ ١٨ ١٧ ١٦ ١٥ ١٤ ١٣ ١٢ ١١ ١٠ ٩ ٨ ٧ ٦ ٥ ٤ ٣ ٢ ١			

September



الاثنين
Monday

21

2015

سبتمبر

١٠ توت ١٤٣٦ ق

٥ ذوالحججة ١٤٣٦ هـ

- ٧ ملخص observable يسع بـ $\text{next}(\text{data})$ عن error حامض مراقبة الـ data
- ٨ http request event تابع لـ real time

٩ لا يستخدم الـ subscribe لـ observable لازم نعمل next / error

- ١٠ اولاً بـ $\text{next}(\text{data})$ الـ data الـ observable ثانياً بـ $\text{error}(\text{error})$ الـ error الـ observable ثالثاً

١١ ~~الـ data الـ observable ينبع من $\text{next}(\text{data})$ الـ data الـ observable ينبع من $\text{error}(\text{error})$ الـ error الـ observable~~

- الـ functions لـ subscribe لـ observable
- ١٢ $\text{data}(\text{next}(\text{data}))$ لـ $\text{next}(\text{data})$ الـ data الـ observable لـ $\text{next}(\text{function})$ الـ function الـ observable $\text{error}(\text{error})$ الـ error الـ observable $\text{complete}(\text{complete})$ الـ complete الـ observable

١٣ دو الـ RxJS و ReactiveX.js

- ١٤ عبارة من javascript library يستعمل مع الـ observable بـ operator او api يستعمل مع الـ observable بـ operator مع الـ asynchronous code او $\text{observer design pattern}$ الـ library

١٥ الـ observable هو مراقبة الـ data في زمن.

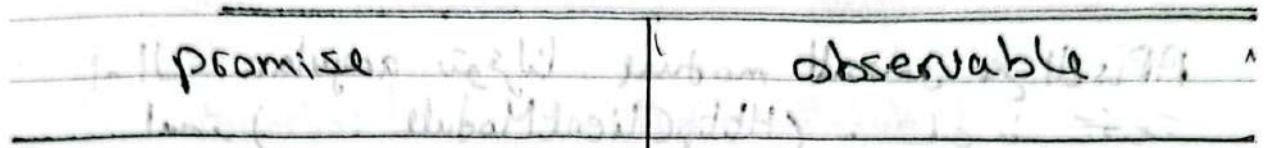
١٦ ~~الـ data الـ observable ينبع من $\text{next}(\text{data})$ الـ data الـ observable ينبع من $\text{error}(\text{error})$ الـ error الـ observable~~

September

Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

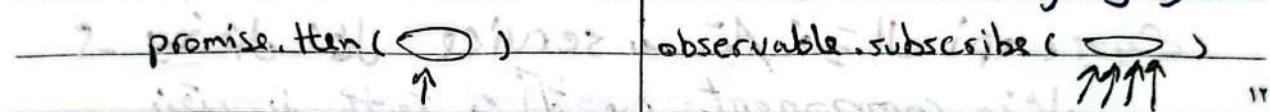


انه هو الفرق بين ال promise وال observable



ال promise يستنفر من وظيفة خط، بينما ال callback function من ال then().

ال observable event observable يصدر بدل call مع اى تغير لبيان ال subscribe() function. ال subscribe() function يمكن تضمينه أكثر من مرة.



ال observable منفرد من توقيفة لا يستنفر.

ال observable operators يوفر observable operators من خارج ال data الا راجعة.

ال lazy operator ي遲ي بثباته execution طول ال creation.

ال lazy observable execute منه يتمثل في subscribe() باستخدام subscribe() method.

الميزة من ال observable انه:

- ① cancellable يمكن الغلق
- ② subscribe يمكن مسحها
- ③ lazy يعني مبكرة

١٩

١٩

السبت

Saturday

2015

سبتمبر

أتوت ١٤٢٢

٦ ذو الحجه ١٤٣٦هـ

② Http Client

١- APIs للتعامل مع التبادل module يتوفر لنا مع angular اسمه (HttpClientModule) في المódule app.module.js من import للزم نعمل

٢- للmodule HttpclientModule من التعامل مع api

٣- يدخل نعمل service في api، بحيث

٤- نقدر نستخدم أكثر من one request واحد فقط

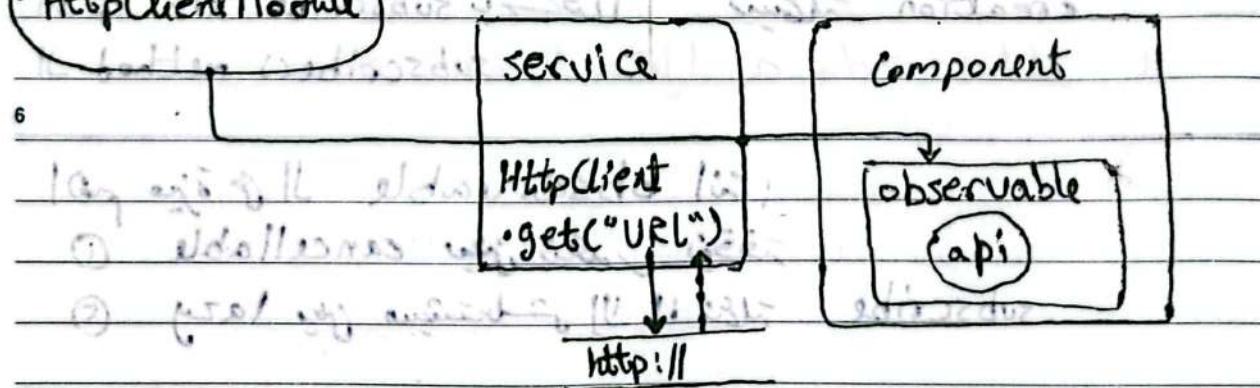
٥- في class يبيّن لـ module

٦- افتر استخداة HttpClient class (HttpClientModule) لـ class

٧- للـ Observable يبيّن api من نوع promise

٨- والـ Observable هو بديل للـ asynchronous code

٩ HttpClientModule



September

Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Sat	Sun	Mon	Tue	Wed	Sat	Sun	Mon	Tue	Wed
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

(1) app.module.ts

```
import {HttpClientModule} from "@angular/common/http";
=
imports: [—, —, HttpClientModule]
```

(2) test.service.ts

```
import {HttpClient} from "@angular/common/http";
import {Observable} from "rxjs";
=
export class testService {
  constructor(private _httpClient: HttpClient) {}
  getAPI(): Observable<any> {
    return this._httpClient.get(' ');
  }
}
```

(3) home.component.ts

```
import {testService} from "../test.service"
export class HomeComponent implements —, —, — {
  constructor(private test: testService) {}
  ngOnInit() { this.getAllData() }
  getAllData() {
    this.test.getAPI().subscribe(
      (res) => console.log(res),
      (err) => console.log(err),
      () => console.log("completed")
    )
  }
}
```

سبتمبر

٦ ثور ١٤٢٢

٤ ذوالحججة ١٤٣٦

```

7   getAllData() {
8     this._test.getAPI().subscribe((res) => {
9       callback <-- {
10      "functions": {
11        "x": res.result,
12        "e": () => {},
13        "c": () => {}
14      }
15    })
16  }

```

```

12 ngOnDestroy() {
13   this._test.getAPI().unsubscribe();
14 }

```

مقدمة الـ observable ، نحن نلقي على امباراً ، يرجع

property / variable (من api)

ex) ~~x~~ x: any;

property من x = this._test.getAPI();

variable من this.x.subscribe((r) => {}, (e) => {}, (c) => {});

this.x.unsubscribe();

first() last() take() takeLast() skip() skipLast()

① اول خطوة اجل module: import { HttpClientModule } from '@angular/common/http';

② ثاني خطوة اجل service import { HttpClient } from '@angular/common/http';

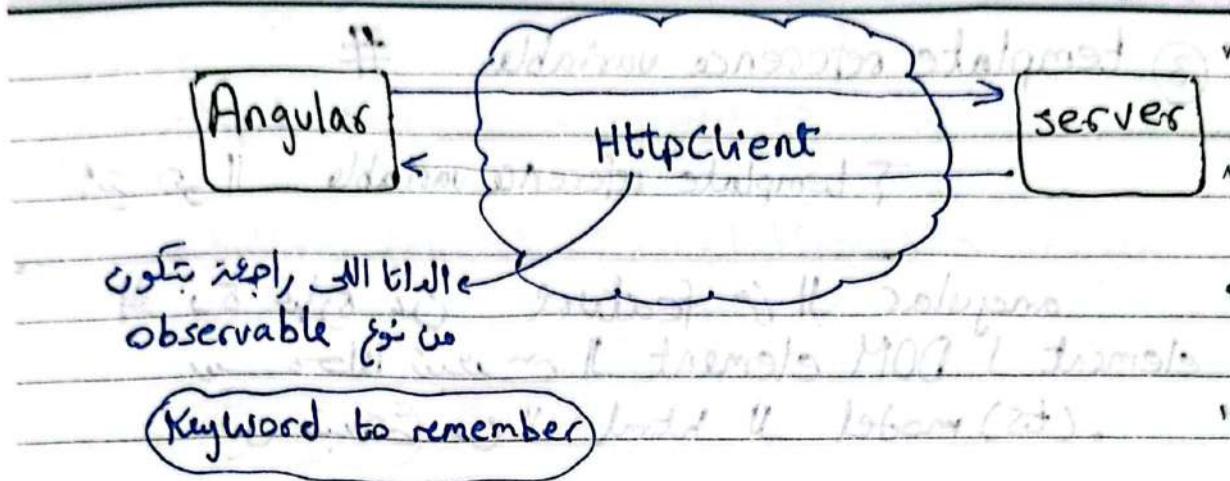
~ ~ ~ Observable<any>

اولاً observable دو method

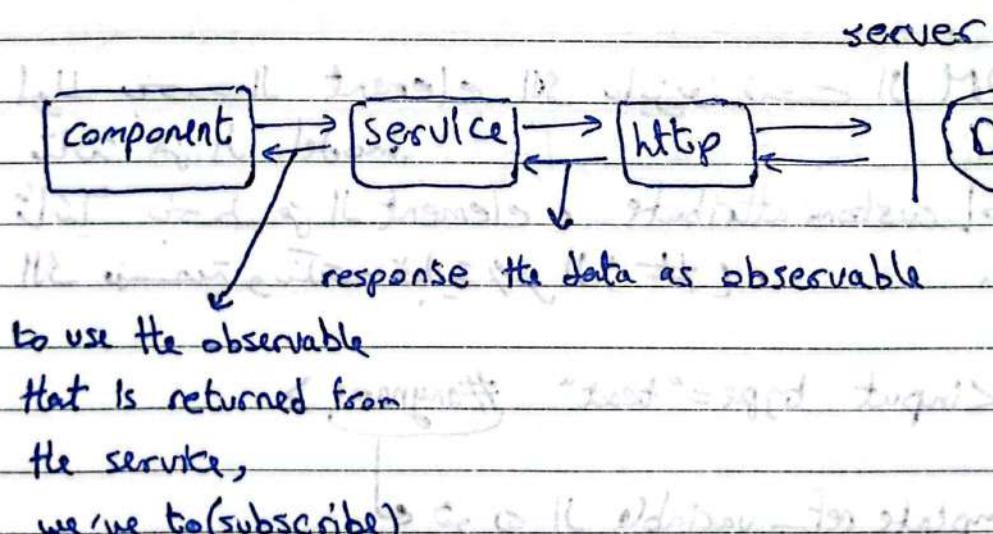
ـ (method) api

ـ component اسخدم service

September		Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18



- 1- HttpClientModule
- 2- HttpClient
- 3- Observable
- 4- .get('URL')
- 5- .subscribe()
- 6- .unsubscribe()



10

15

الثلاثاء

Tuesday

2015

سبتمبر

٤ سبتمبر ١٤٣٢

٢ ذي الحجة ١٤٣٢

٧ ③ template reference variable #

٨ . template reference variable #

٩ # ركّز عبارتك من ال feature

element لـ DOM element

١٠ . (ts) model من html لا مطابق ترتيبه

١١ methods properties وال بالناك تقدر ترجعهم الـ

أيامه بـ DOM في ملف ts

١٢

.js object لـ html element لـ DOM

اولا ينصر الـ element الذي يأخذون بحسب الـ

٣ . ترتيبه من الـ model

ثانياً يبحث عن الـ element احنا

الـ model وينتسب قبلية بـ (#)

٥ <input type="text" #anyname>

٦ template ref-variable هو دو دو الـ reference variable هو مطابق الـ template

September

Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Sat	Sun	Mon	Tue	Wed
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

(EX)

```
<input type="text" #txtInput>
```

```
<button (click) = "m1(txtInput)"> click here
</button>
```

(in ts file)

```
msg = ""
```

```
m1(input){ console.log("msg", msg); }
```

```
console.log(input);
```

```
console.log(input.value);
```

```
this.msg = input.value;
```

```
<p #hmp020><p>
```

متغير hmp020 في view هو variable في dom

الغاية هنا أن إجها نبتت value من ال view لل model.

سبتمبر

٢٠١٧٢٢ تقويم

٢١ ذي القعده ١٤٣٦ هـ

④ pipes

- **pipes** في angular هو feature ذو دلالة على transformation data
- ال purpose (data) بـ **تحويل** معين

10 من هنا نوعين من ال pipes

- 11 (defined pipes) angular يرجع ال pipes ①
 (custom pipes) إنشاء بـ **function** pipes ②

12 **pipe** هو الـ function الذي يحوي على **value** ويعمل عليه **transform** معين.

1 {{ x | lowercase }}

2 {{ x | uppercase }}

3 {{ x | titlecase }}

4 {{ x | slice:3 }}

5 {{ x | slice:4:7 }}

6 {{ 3.657 | number: '1.2-3' }} → 3.657

7 {{ 3.657 | number: '4.4-5' }} → 0,003.6570

8 {{ 3.657 | number: '4.1-2' }} → 0003.65

custom pipe

ازل نعمل

1- using pipe square لم

2- square.pipe.ts بتحوّل create pipe او transform() اسم function

@Pipe() .pipe() يتابع ال logic

.name: 'square'

}) class

export { SquarePipe } implements PipeTransform {

transform (value: number): unknown {

return value * value;

}

3- in html

{{ 10 | square }} → 100

(محوّلة) اسم ال custom pipe
 لدوم تكون unique وسـم مـنـكـرـرـ مـنـ
 (angular defined pipes)
ولـا مـنـهـ مـيـتـلـ

الثلاثاء	الأربعاء	الخميس	الجمعة	السبت	الأحد	الاثنين	الثلاثاء	الأربعاء	الخميس	الجمعة	السبت	الأحد	الاثنين	الثلاثاء	الأربعاء
٣٠	٢٩	٢٨	٢٧	٢٦	٢٥	٢٤	٢٣	٢٢	٢١	٢٠	١٩	١٨	١٧	١٦	١٥

September

سبت

السبت ١٧٢١ في

٢٧ ذوالقعدة ١٤٣٦ھ

7. ١٦. ٦٣. ؟ parameter بـ يساوي pipe لا (زائر) (Q)

8. transform (value: number, v2: number): number {
return value * v2;

9. } { ١٥ | square: ٥ } → ٥٠

10. ١٥ | square: ٥ → ٥٠

11. parameter الـ ١٥ | square: ٥ parameter الـ ٥ → ٥٠

12. الأول الثاني

1. transform (value: number): number {
return value * value;

2.

3.

4.

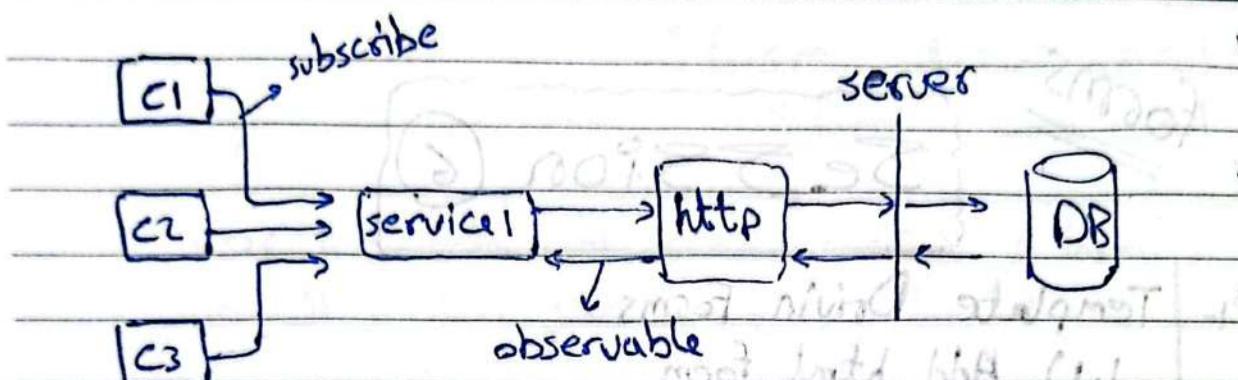
٥٠ → Prompt of ٥

5.

6.

September

Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Sat	Sun	Mon	Tue	Wed
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21



١- الـ component

الـ http request ينبع من service

الـ server ينبع من http request

الـ DB ينبع من server

الـ http request ينبع من DB

الـ observable object ينبع من data باـ http

الـ subscribe هيكل نظمته لـ observable

9

9

الأربعاء

Wednesday

2015

سبتاء

٤ نيسان ٢٠١٥

٢٥ ذي القعدة ١٤٣٦ هـ

~~forms~~

Session 6

1- Template Driven Forms

- 1.1) Add html form.
- 1.2) Bind data.
- 1.3) track state & validity.
- 1.4) provide visual feedback.
- 1.5) display error messages
- 1.6) posting data to the server.

2- Reactive Forms

- 2.1) add html form.
- 2.2) create form model.
- 2.3) manage form control values.
- 2.4) FormBuilder service
- 2.5) validation → simple
 - custom
 - cross-field
 - dynamic
- 2.6) add dynamic form controls.
- 2.7) submit form data.

September

Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Sat	Sun	Mon	Tue	Wed
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

ابدأ من الـ form
امثلة على الـ form

الـ form هي طريقة بسيطة لمحاجنا ناشر data من الـ user.
هذا ينبع من الـ form لم يتمكن من data من الـ user.

لما زلنا نتعامل مع الـ forms في الـ Angular

الـ forms لها طرقين للتتعامل مع الـ angular

Template Driven Forms ①

(Reactive Forms) Model Driven Forms ②

Reactive Forms

→ heavy on the component class
الـ logic آليات في الـ ts

1- لا يوجد خالية في الـ ts
2- يستحسن التعامل مع
complex scenarios

custom validation

Template Driven Forms

→ heavy on the component html
يسهل الـ logic بينها وبين الـ html

1- الآلور موجود في الـ html
2- سهل الاستخدام
easy to use
two way binding
[ngModel]

3- يتحقق من الـ validity state والـ track
4- يستحسن مع الـ simple forms
5- يستحسن لـ unit test

V

7

الاثنين
Monday

2015

سبتاء

٢٠١٧٢١ نسخة

٢٣ ذوالقعدة ١٤٣٦ هـ

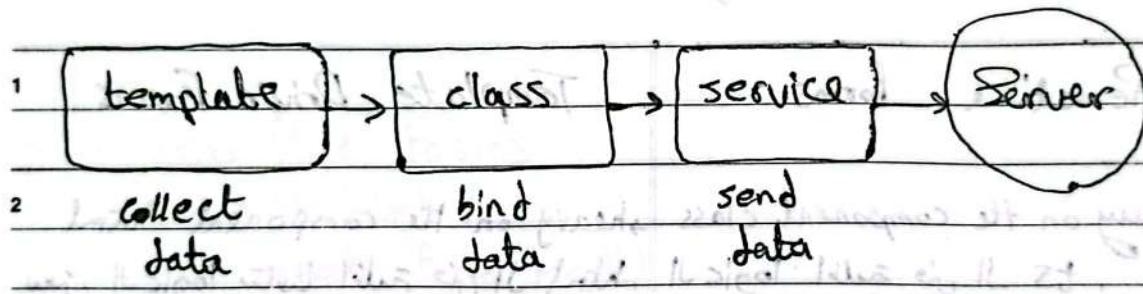
Angular Forms

① the component template (html) contains HTML to collect the user data.

② the component class handling data binding, which means it takes the data from the view.

③ the collected data will be sent from the component class to the server via service.

12



Q) أيهما الجيد بين الـ

& Reactive Forms

5. html لـ template driven forms

6. TS لـ Model driven forms

7. فـ forms مع الـ template driven forms في unit testing وـ register وـ login

September

Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

الأحد
ابعدت الـ data لـ السيرفر .
الـ form داخل ①
data ـ ② خارج من ③ المفهوم ④

٢٠١٥

١٧٢١ في

٢٢ ذو القعده ١٤٢٦ هـ

سبتمبر

① adding the html form :

(ex)

```
<input type="text">
<input type="email">
<input type="tel">
```

② binding the data [ngModel], ngForm

لـ template driven forms يتعين طريقة الـ ①

2-way binding بالـ ② نقرر تـ ③ (Forms Module) الـ import ④

template driven forms need use 2-way binding [ngModel]

أول خطوة من الـ ① وقتها ستـ ② form الـ ③

ـ ④ (ngForm) جـ ⑤ use directive.

زيـ ⑥ عن الـ form زـ ⑦ "value" وـ ⑧ input

وـ ⑨ valid ولـ ⑩ المـ

ـ ⑪ نوصل مـ ⑫ عن الـ ⑬ template reference variable

<form #userform="ngForm">

ـ ⑭ variable

ـ ⑮ مـ ⑯ form element

ـ ⑰ مـ ⑱ inputs الـ ⑲ جـ ⑳

</form>

٥

٥

السبت

Saturday

2015

سبتمبر

٢٠١٧/١٧/٢٠١٣

٢١ ذوالقعدة ١٤٣٦ھ

٧

- ٧ خطاً خطوة باتابة value أو trackBy ngForm بجانب الـ input لـ input معين ، لازم الـ input يتوفر فيه . name يحتوى على سترطين ① يحتوى على . ngModel ② يحتوى على

٨ <form #userForm = "ngForm">
 ٩ <input type="text" name="Name" ngModel>
 ١٠ <input type="email" name="Email" ngModel>
 ١١ <input type="tel" name="phone" ngModel>

١٢ </form>

١٣ ngForm بجانب الـ input يحصل على binding لـ name يحتوى trackBy

١٤

- ١٥ خطاً خطوة باتابة group of inputs لـ input name "Name" ngModel
 ١٦ <ngForm> بجانب الـ input لـ object name "Name" ngModelGroup = ""

١٧

١٨ <form #userForm = "ngForm">
 ١٩ <input type="text" name="Name" ngModel>
 ٢٠ <div ngModelGroup="address">
 ٢١ <input type="text" name="street" ngModel>
 ٢٢ <input type="text" name="state" ngModel>

٢٣ </div>

٢٤ </form>

User

Name = "Ali"

address street = ""

state = ""

September

Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed
١	٢	٣	٤	٥	٦	٧	٨	٩	١٠	١١	١٢	١٣	١٤	١٥	١٦

رابع خطوه دخل class نجمع فيه كل الـ fields من object ودخل من الـ form binding

>> ng g class User

export class User {

public

constructor (public name:string,
public email:string,
public phone:string,
public address:string,

=) { }

}

تل field من

الباري عن ملحوظ

هذين في الـ value

تباعه input

خامس خطوه عرف object من الـ form على الـ .form

userModel = new User ('', '', '', '')

سادساً اعمل bind بين الـ inputs والـ .ngModel

<input type="text" name="email" [(ngModel)]="userModel.email">

③ Tracking state of validity

- كثير input / form control لـ by default بـ مجموعه directives classes
- all state لـ بـ "state" لـ "visited"

state	if true	if not
control has been visited.	ng-touched	ng-untouched
control value has been changed	ng-dirty	ng-pristine
control value is valid.	ng-valid	ng-invalid

لو الـ focus من مجموعه control
 لو الـ value لـ control
 لو الـ validation لـ control

دول عباره عن classes ينطوي على
 input مثلاً: يضاف class لـ "test"
 وكل class

6 (ex)

```
<input type="text" name="name" class="test">
<p>{{name.className}}</p>
```

→ "test ng-untouched ng-pristine ng-valid"

٧. إنّه هو البديل للـ classes دينـ لـ classes دينـ لـ classes دينـ لـ classes دينـ

يـعنـى طـريقـى الـ ngModel دـيـنـ لـ properties بـتـوحـى الـ ngModel دـيـنـ وـيـنـعـى الـ اـلـاسـمـ الـ تـصـيـرـ

ng-touched \Rightarrow touched properties

ng-unouched \Rightarrow untouched ~

ng-dirty \Rightarrow dirty ~

ng-pristine \Rightarrow pristine ~

ng-valid \Rightarrow valid ~

ng-invalid \Rightarrow invalid ~

٨. t-r-v نـقـرـنـوـصلـ لـ properties دـيـنـ خـارـجـىـ جـواـ

<form #userForm = "ngForm">

<input type="text" #name = "ngModel" />

properties يـعـتـقـدـ عـنـ ngModel

. control يـتـالـخـرـجـ الـ clـasـsesـ بـالـ

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

<p>{{name.touched}}</p> } false او true

</form>

٩. html element لـ نـقـرـنـوـصلـ لـ مـعـلـومـاتـ مـنـ الـ notes

(template reference variable) لـزـمـ سـتـخـتـمـ انـ

وـهـ عـارـفـ عـنـ variable يـحـتـوىـ عـلـىـ مـعـلـومـاتـ عـنـ

الـ element

سبتمبر

٢٦ مسري ١٤٣١ق

١٧ ذوالقعدة ١٤٣٦هـ

- ٧ ④ providing visual feedback ادى علیه style
- ٨ لا visual feedback هو ان احنا نرى علامات validation ال بين ال
- ٩ الـ validation هو ان احنا نعرض رسالة display msg for feedback ←
- ١٠ validation . validation ال بين ال
- ١١ validation الـ (اجعلها) لـ visual feedback ← توضح اى عرض
- ١٢ validation لـ (اجعلها) لـ validation الـ (اجعلها) توضح اى عرض
- ١ display message يبيّن اى عرض
- ٢ input الـ style ادى c invalid value لـ value (ex) خاتمة لونه اخضر.
- ٣ ex <input type="text" class="t" name="name" ngModel>
- ٤ color: red; لبعض المظاہرات / لبعض الخواص
- ٥ border: 1px solid red; border لـ element
- ٦ (ex) <input type="text" class="t" name="name" ngModel> ←
class مرتبط ← (t) اسمه ← (t) ← لـ

امض ال `input` focus بـ `true` لو على `validation` ال `value` حقق ال `valid`
 امض ال `input` dirty. سـ `dirty` يعني انه خارج `input` .

اذا ندخل `input` مع ال `validation rules` فـ `dirty` يتحقق من ال `validation` من `html` من `ts`

عن طريق استخدام ال `HTML property` `pattern` من `html` من `regex` .
`, pattern = " ^ [0-9]{11} $ "`

`(cx) <input pattern = " ^ [0-9]{11} $ " --->`
`RegEx`

`<input type = "text" value = "11111111111" />`
`<input type = "text" value = "11111111111" />`

أغسطس

٢٤ مسرى ١٤٣٦ ق

١٥ ذوالقعدة ١٤٣٦ هـ

٧ ⑤ display error message

٨ **ما هي طريقة من ظهور معيّن بـ "element"؟**
لذلك نكتب . (ngIf) condition

٩

`<input ----- #name="ngModel" ----- />`

١٠ **الآن نكتب "if" condition**

`<p [ngIf="name.invalid || name.unouched">`

١١ **Here is error here**

`</p>`

١٢ **(٥)**

`<p [class.d-none] = "name.invalid">`

`</p>`

٢ **إذاً نعمل مجموعة مختلفة لـ "error messages"؟**
الـ "error" يرجع إلى الكـ "element"

٣

`<input ----- #name="ngModel" ----- />`

٤ `<div "ngIf = "name.errors && (name.invalid || name.touched)">`

`<p [class.d-none] = "name.valid"> ----- </p>`

٥

`<p [class.d-none] = "name.dirty"> ----- </p>`

٦

`<p "ngIf = "name.errors.required"> ----- </p>`

٧

`<p "ngIf = "name.errors.pattern"> ----- </p>`

`</div>`

August

Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

السبت

Saturday

٢٩

٢٠١٥

٢٢ مسري ١٤٣٦ ق

١٤ ذو القعدة ١٤٣٦ هـ

أغسطس

إذاً نحتاج الى button يكون disabled
حلول ما فيه حاصل invalid

```
<form #userForm="ngForm">  
    <input type="submit" [disabled] = "userForm.form.invalid">  
</form>
```

إذاً ندخل في browser تجرب validation
إذاً ندخل تأشير form في browser

```
<form #userForm="ngForm" novalidate>  
    </form>
```

validation property هذه
الخاصية بالـ browser
بيان validation من حيث
بيان .

السبت	الأحد	الاثنين	الثلاثاء	الأربعاء	الخميس	الجمعة	السبت	الأحد	الاثنين	الثلاثاء	الأربعاء	الخميس	الجمعة	السبت	الأحد	الاثنين	الثلاثاء	الأربعاء	الخميس	الجمعة	السبت	الأحد	الاثنين	الثلاثاء	الأربعاء	الخميس	الجمعة	السبت
٢٩	٣٠	٣١	٣٢	٣٣	٣٤	٣٥	٣٦	٣٧	٣٨	٣٩	٤٠	٤١	٤٢	٤٣	٤٤	٤٥	٤٦	٤٧	٤٨	٤٩	٥٠	٥١	٥٢	٥٣	٥٤	٥٥		

August

أغسطس

٢٢ مسرى ١٤٣١ق

١٣ ذوالقعدة ١٤٣٦ھ

٦ send the data in form to server

٧ event binding من طريق الـ `(ngSubmit)` ← الى form بالـ `event` فيه

٨ ① <form _____ (ngSubmit)="sendData()">

٩ </form>

١٠ ② userModel = new User(' ', ' ', ' ', ' ');

١١ sendData() {
this.userService.enroll(this.userModel).subscribe();

١٢ } } ~~for (let i = 0; i < 10; i++) {~~

١٣ ③ enroll(user:User) {

١٤ return this.httpClient.post<any>(`http://localhost:3001/api/users`, user);

١٥ } } ~~for (let i = 0; i < 10; i++) {~~

١٦ هذا يرجع بـ observable
١٧ subscribe لـ observable للـ observable

١٨ لذلك نعم

١٩ لذلك نعم

٢٠ لذلك نعم

٢١ لذلك نعم

٢٢ لذلك نعم

٢٣ لذلك نعم

٢٤ لذلك نعم

٢٥ لذلك نعم

٢٦ لذلك نعم

٢٧ لذلك نعم

٢٨ لذلك نعم

٢٩ لذلك نعم

٣٠ لذلك نعم

٣١ لذلك نعم

August

Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

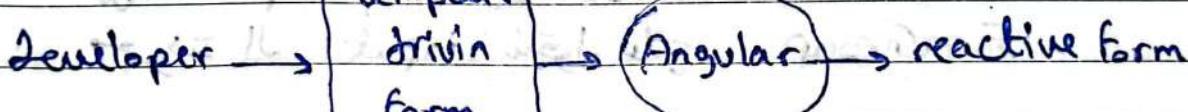


محظوظ

- ١- الـ **(FormGroup)** ينكرى من نوع **form**
- ٢- الـ **(FormControl)** ينكرى من نوع **input**

٣- الـ **template driven forms** يستعمل مع الـ angular internally من طرحي الـ reactive form . behind the scene

الـ **template form** يجعل الـ angular "انتـر فـارم" وراء الكواليس



and in real not with angular program (electromagnetism).



Reactive Forms

- complex scenarios \rightarrow better suited for reactive forms
- component's class \rightarrow all logic

① Add HTML form

`<form>` \rightarrow formGroup in angular

- 11 `<input type="text">` \rightarrow FormControl in angular
- 12 `<input type="email">` \rightarrow FormControl in angular
- 13 `<input type="password">` \rightarrow FormControl in angular
- `</form>`

• FormControl is instance of input

• FormControl is input

• FormGroup is instance of input

3

```
name: FormControl  
email: FormControl  
pass: FormControl
```

FormGroup

• Reactive form \rightarrow import `(ReactiveFormsModule)`

angular للـ form او

كلـ formGroup → form الـ third item

Tuesday

25

٢٥

٢٠١٥

١٤٣٦ هـ

١٧٢١ مـ

أغسطس

② create Form model

الـ FormGroup هو عبارة عن form model يحتوى على FormControls

الـ typescript هو محاكى للـ form model . ts code

```
registerForm = new FormGroup({  
    userName: new FormControl(''),  
    email: new FormControl(''),  
    pass1: new FormControl(''),  
    pass2: new FormControl('')  
});
```

(registerForm) ← انتـ Form model

③ binding b/w html form & form model :-

```
<form [FormGroup] = "registerForm">  
    <input type="text" formControlName = "userName" />  
    <input type="email" _____ = "email" />  
    <input type="password" _____ = "pass1" />  
    <input type="password" _____ = "pass2" />  
</form>
```

أغسطس

١٨ مصري ١٤٣٦

٥ ذوالقعدة ١٤٣٦

٧ nesting Form Groups

٨ Form Group بداخل Form Group ← هي مبارقة عن بعدها يفتح الـ Form Group الباقي ←

٩

(ex)

١٠ registerForm = new FormGroup({})

userName: new FormControl(''),

email: new FormControl(''),

address: new FormGroup({}),

state: new FormControl(''),

city: new FormControl('')

}),

}),

٢

<form [formGroup]= "registerForm" >

<input type="text" formControlName = "userName" />

<input type="email" formControlName = "email" />

٤

<div formGroupName = "address" >

<input formControlName = "state" />

<input formControlName = "city" />

٦

</div>

</form>

{{ registerForm.value | json }}



- Q) ازاي ندخل ازرار الى form
 Q) ازاي ندخل data فى ال form من API
 Q) ازاي ندخل API فى data من form

عن طريق setValue() امتحن بياخد
 value: الاسم فى object وال value جواك

```
<button type="submit" (click)="getDataFromAPI()">  

</button>
```

```
getDataFromAPI() {  

  this.registerForm.setValue({  

    userName: "Amir",  

    email: "@-",  

    address: {  

      state: 'egypt',  

      city: 'egy'  

    }  

  });
```

initialization لـ it تختلف بين setValue() و patchValue() لـ form model

لـ inputs لـ initialization متى كل الـ inputs بالتأديب

patchValue() من setValue()

④ FormBuilder service

بشكل form model لـ service الـ
أبسط / مبكر "أقل".

form model ~~لـ~~ يرتبط بـ service ما من إلا

(ex) ~~form builder's~~ ~~methods~~
in dependency injection ~~عبر services~~ أو
طريق الـ constructor

constructor(private fb: FormBuilder) { }

registerForm = this.fb.group({

name: ['',],

email: ['',],

address: this.fb.group({

state: ['',],

city: ['',]

})

})

من الآخر،

this.fb.group() ← new FormGroup()
[] ← new FormControl()

group() هي عبارة عن ~~شيء~~ مـ

FormControl [] الـ FormGroup من instance

٥ Validation

(٥.١) Simple validation

عند بحث عن Validators في class form
will find static properties من جزء

(ex) registerForm = this.fb.group ({
name: ['', Validators.required]
});

① <input [class.t] = "registerForm.get('name').invalid
&& registerForm.get('name').touched" >
.t{
border: 1px solid red;
}

② <p [class.d-none] = "registerForm.get('name').invalid" >
there is an error

</p>

parameter الـ FormControl

ال الاول هو الـ default value

الثاني هو الـ validation

الثالث هو الـ

. array من validation
name: ['', [Validators.required, Validators.minLength(16)]]

٥.

20

الخميس

Thursday

(conditions)

2015

أغسطس

١٤٢١ مصري

١٤٣٦ هـ ذو القعدة

٧. input الـ state او check

٨. registerForm.get('name').valid

.invalid

٩. registerForm.get('name').touched

.untouched

١٠. registerForm.get('name').dirty

١١. validation checks

١٢. registerForm.get('name').required

.minlength

لاروم شارى

عاج

errors

اسم الـ ^{أولاً} _{form model} اسماً _{form control} _{validator}

registerForm.get('name').errors.required

registerForm.get('name').errors.minlength

٤. errors all property

لو محس موجود، او لو رجع بـ null او لـ empty

٥. optional ، يحصل خالية

٦. registerForm.get('name').errors?.required

optional

.required access مخصوص ، اجل errors

[FieldContainer validity={{isValid}}]

August

Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

CS

CamScanner

المسحوة ضوئياً

(5.2) custom validation

يُدخل الـ validation logic إلى ملف ts منفصل
لوجهة reusability.

(ex) `userName.validator.ts`

الـ validator هو الترجمة ما دو الـ function، وداعياً لازم
يأخذ parameter من نوع `AbstractControl`.

```
import { Validator } from '@angular/forms';
import { AbstractControl } from '@angular/forms';

export function forbiddenValidator(control: AbstractControl): {[key: string]: any} {
  const x = /(^([0-9]{15})$)/.test(control.value);
  return x ? { "forbidden": { value: control.value } } : null;
}
```

`registerForm = this.fb.group({
 name: ['', forbiddenValidator]
})`

`<p ngIf="registerForm.get('name').errors?.forbidden"></p>`

ـ الـ parameter ما دو الـ validator
ـ الـ `AbstractControl` من نوع

ـ يرجح عباره من `{ key: value }`
ـ اسم الـ `key` هو اسم الـ `errors` الكـ راجع.

ـ يُدخل أي custom validator إلى
ـ component.

أغسطس

١٧٢١ مصري

١٤٣٦ هـ ذي القعده

٧ regex سـ تعـيل Validator \rightarrow parameter ٥

٨ . function \rightarrow return \rightarrow function

٩ Export function V(x: RegExp): ValidatorFn {

١٠ return (control: AbstractControl) => {

١١ const F = x.test(control.value);

١٢ return f ? { "e": " " } : null;

١ registerForm = this.fb.group({
 name: ['', Vc / ^011\\$ /]]

٢ })

٣ <p *ngIf = "registerForm.get('name').errors?.e" > </p>

٤ (محظوظ)

٥ الـ AbstractControl يورث من الـ formGroup

٦ الـ AbstractControl يورث من الـ formControl

٧ الـ formControl يورث من الـ validator

٨ الـ Validator يورث من الـ ControlValidator

٩ لكن لو أكتبنا الـ validators من الـ formGroup

١٠ فـ validators يورث من الـ formGroup

١١ فـ validators يورث من الـ formGroup

August

Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

(5.3) cross-field validation

ما صوراً لـ custom validator، لكن يمكن عمل مستوى الـ formGroup.

form group يختلف عن مستوى الـ cross field validator، وليس مع مستوى الـ controller.

```
export function ConfirmValidator(controls: AbstractControl) {
  const pass1 = controls.get('pass1');
  const pass2 = controls.get('pass2');

  return (pass1 || pass2) ? { 'misMatch': true } : null;
}
```

```
registerForm = this.fb.group({
  pass1: ['', [ConfirmValidator]],
  pass2: ['']
}, { validators: [ConfirmValidator] });

```

```
<p *ngIf="registerForm.errors?.misMatch"> </p>
```

أغسطس

١٧٢١ مصري

١٤٣٦هـ ذوالقعدة

(5.4) dynamic validation

* يمكن تصريف الـ validation لـ "runtime" معين في الـ condition

9

subscribe to checkbox (مثال) لو ضبطت checkbox required Validator يتحقق التصريف

```

1 <input type="email" formControlName="email">
2 <input type="checkbox" formControlName="subscribe"
3   (click) = "setEmailValidation" >
```

```

1 registerForm = this.fb.group({
2   email: ['', ],
3   subscribe: [false]
4 })
```

```

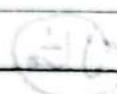
5 setEmailValidation() {
6   this.registerForm.get('subscribe').valueChanges.subscribe(
7     (value) => {
8       if (value) {
9         this.registerForm.get('email').setValidators([
10           validators.required
11         ]);
12       } else {
13         this.registerForm.get('email').clearValidators();
14       }
15     });
16 }
```

August

Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	25	26	27

لـ `updateValueAndValidity()`

لو علمنا تغيير الـ `validator` من الـ `runtime` في `controller` لازم نعمل `updateValueAndValidity` بار الجرس

 In bad situations, you can't

use super.initState and await setState.

١٨٢٣ as due to the framework's "Wasm" form
until it's ported to webAssembly, we'll continue

Diagnostic code + modifications coming
[!] Fixes

(1) `your_widget.addPostFrameCallback`

as

function as

as `yourWidget.addPostFrameCallback((Duration duration) {`

`final GlobalKey<YourWidget> key = GlobalKey<YourWidget>();
<code>final GlobalKey<YourWidget> key = GlobalKey<YourWidget>();</code>`

"`key.currentState!.validate()`" `key.currentState!.validate()` will

<"`key.currentState!.validate() == true`"> `return`

<"`key.currentState!.validate() == true`"> `return`

Aug

أغسطس

٢٠١٥ مصري ١٧٣١

١٤٣٦ هـ ٢١ شوال

⑥ Add dynamic form controls

فـ input / formControl في نقر سيف يـ جـيـزـ فيـ الـ runtime

ـ اـ جـاـلـ جـاـلـ teacher[] اـ جـاـلـ جـاـلـ اـ جـاـلـ جـاـلـ

ـ اـ جـاـلـ جـاـلـ اـ جـاـلـ جـاـلـ

this.fb.array() ← FormArray طـرـيقـ الـ وـزـارـةـ اـ جـاـلـ جـاـلـ

ـ runtime اـ جـاـلـ جـاـلـ اـ جـاـلـ جـاـلـ

① ex registerForm = this.fb.group({

email: ['',],

addEmailField: this.fb.array([])

});

② get addEmail() {

return this.registerForm.get('addEmailField') as FormArray;

}

addNewEmail() { this.addEmail.push(this.fb.control('')) }

③ <button (click) = "addNewEmail()"> add </button>

<div formArrayName = "addEmailField" "addEmailField"

"ngFor = "let email of addEmail.controls; index as i" >

<input type = "email" [FormControlName] = "i" >

</div>

August

Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

$\left\{ \begin{array}{l} [\text{FormControlName}] = "1" \\ [\text{FormControlName}] = "email" \end{array} \right.$ لَهُ عَلَنَا !
 مُثْرِّي

هنا نقدر نكّد بـ بـ مُولَّة عن طريق الـ index
 (ex) addField([0]) [١]
 وَكَذَا ...

اَهَى نَحْنُ الـ FormArray
 لَا نَتَوَزَّعُ نَحْنُ control اَو repeat اَو معِنْ ، او
 اَم formGroup

ir

12

الأربعاء

Wednesday

2015

أغسطس

٦ مஸرى ١٧٣١ ق

٢٧ شوال ١٤٣٦ هـ

7

Session ٧

8

٩ ١- Authentication & Authorization

١٠ ٢- Jint

١١ ٣- routing guard

١٢ ٤- BehaviorSubject

١٣ ٥- Route parameter

١٤ ٦- search pipe

١٥ ٧- lazy loading

١٦ ٨- interceptor

١٧ ٩- production

6

August

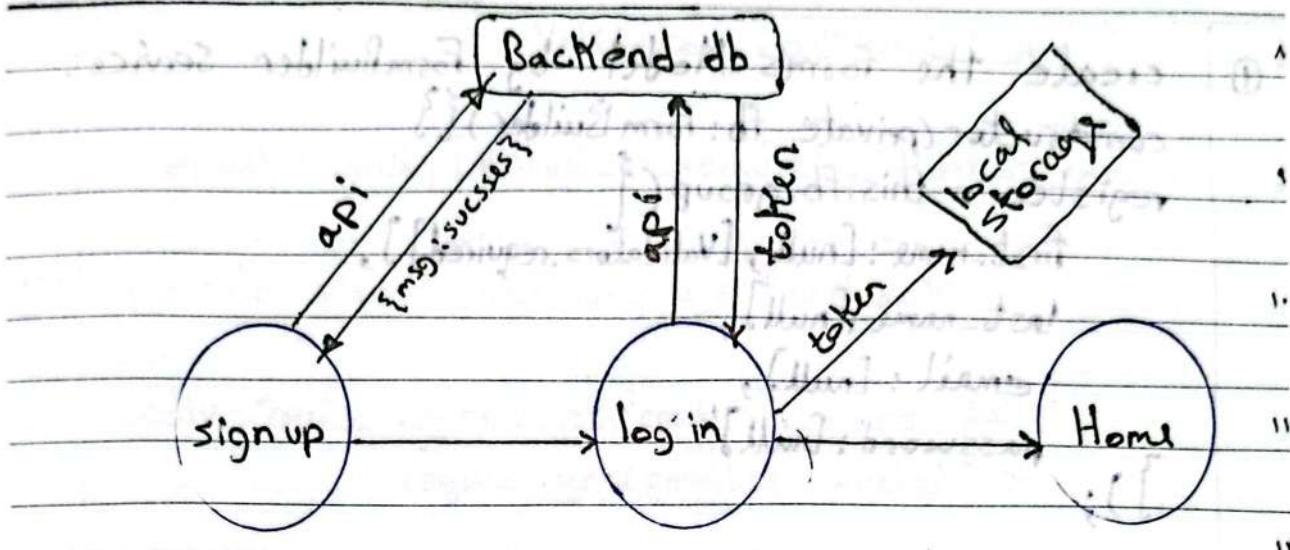
Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24



CamScanner

المسوحة ضوئيا بـ

① Authentication



complex forms like Reactive forms
registration or business logic

create sign up & login forms steps

- 1- add html form in html.
- 2- create the form model in ts.
- 3- bind b/w html form and form model.
- 4- add the validation
- 5- provide the visual feedback and error msgs.
- 6- post the data to the Server Via the API.
- 7-
at registration → navigate to login page.
at login → navigate to home page and provide the authorization by using the Token in the localStorage.

أغسطس

٤ مسري ١٤٣٦ ق

٢٥ شوال ١٤٣٦ هـ

* registration form

- ① create the form's Model by FormBuilder Service.

```
constructor(private fb: FormBuilder){}
register = this.fb.group({
  first_name: [null, [Validators.required]],
  last_name: [null],
  email: [null],
  password: [null]
});
```

- ② add the html form

```
<form>
  <input type="text">
  <input type="email">
</form>
```

- ③ bind b/w html form & form Model.

- <form [formGroup]="register">

- <input formControlName="first_name">

- <button (click)="sendData()" -->

(on) if it is not at proxy

- <form (ngSubmit)="sendData()" -->

August

Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

(٧) add Validation

password: [null, [Validators.required,

Validators.pattern('^[A-Z]{6}{10}\$')]]

email: [null, [Validators.required, Validators.email]]]

(٨) provide visual and message feedback

<div *ngIf="register.get('email')?.errors &&

register.get('email')?.touched">

<p *ngIf="register.get('email')?.errors.required></p>

<p *ngIf="register.get('email')?.errors.email></p>

<div>

(٩) post user data / integrate with auth APIs

(cat) auth.service.ts

constructor(private h: HttpClient){}

register(data: any): Observable<any>{

return this.h.post('URL', data);

}

(cat) register.component.ts

sendData(){

if(this.register.invalid){ return; }

else{ this.authService.register(this.register.value).subscribe(

(data)=>{if(data.message == 'success'){ }}

else{ }}

}); } ↳ error

↳ navigate to

٨

السبت

٨

Saturday

2015

أغسطس

٢٠١٣ ميلادي

١٤٣٦ هـ

٧ ② JWT → json web token

٨ آية دو ال authorization وال authentication (٢)

٩ ال authentication هو عملية تسجيل الدخول user حتى الموقع من طريقة registration او login.

١٠ ١١ ال authorization هو ما بعد login ، هو ال administration

١٢ ال user ! انه يتحقق و اية لدك

Authentication - Authorization

- login - what permissions
- who are you? do you have?

١٣ ١٤ آية انا اذكر ان ال user متى جعل عندي authentication

١٥ ١٦ ال authorization هو الصلاحيات التي ال user يقرر بخلاف

١٧ ١٨ آية ازاي تثارد ال login user على login (٢)

١٩ ٢٠ عن طريق ال token في response login request

٢١ ٢٢ token هو عبارة عن معلومات ال user على login ولكن مسحوبة و المطلوبات فيه تكون معلومات user يعني مطحوناً غير ملحوظة تنظر بامان ال user insensitive

August

Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon
١	٢	٣	٤	٥	٦	٧	٨	٩	١٠	١١	١٢	١٣	١٤	١٥	١٦	١٧	١٨	١٩	٢٠	٢١	٢٥	٢٦	٢٧



CamScanner

المسحقة ضوئياً



Q) ايه هو ال token وبيكون من ايه؟

1- ال token هو عباره عن معلومات معينه عن ال user الـ login، لكن المعلومات ديره بيكون من غيره.

2- خاصيه token انه يعرفن كل ال user على login ولا لا، ومن خلال معلومات ال user من ال token، اقدر اقدر صلاحياته على user / authorization

3- ال token بيكون من ٣ أجزاء:

اول جزء هو ال Header وهو عباره عن ال algorithm وال secret key

ثاني جزء هو ال payload وهو عباره عن ال data بتاعة ال user

ثالث جزء هو ال signature وهو عباره ناتج تشفير ال payload وHeader بال base64 ومضاف السرور secret key، قيمة secret key بيكون موجوده في ال server فقط، بنستخدم secret key لـ create token ونجعله لل user ونطلع على data من secret key.

$$\text{signature} = (\text{Header} + \text{payload}) \text{base64} + \text{secret}$$

٧

6

الخميس

Thursday

2015

أغسطس

١٧٢١ أبيب

٢١ شوال ١٤٣٦ هـ

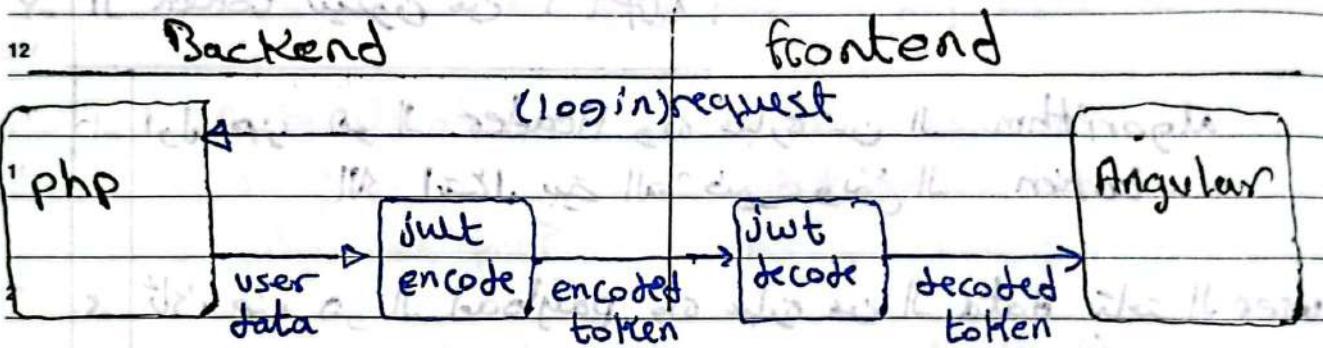
٧) ما هو ال jwt ؟

٨) ال jwt عبارة عن library يستعمل backend مكان برمجوا تغير data بتاعة ال user

٩) واحنا ك front end نعمل بناء على ذلك تغير data

١٠) ال jwt اتبيس من backend token

١١) ال jwt هو مكتبة يستعمل بناء على encode و decode



١٢) npm i jwt-decode

١٣) from import jwtDecode from "jwt-decode"

١٤) userData: any = null;

١٥) saveUserData()

```
let encodedToken = JSON.Stringify(localStorage.getItem('token'));
```

١٦) let decodedToken = jwtDecode(encodedToken);

this.userData = decodedToken;

}

August

Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	25	26	27

CS

CamScanner

المسوحة ضوئياً

③ guard ➞ ng g g auth ↪ guard

interfaces كما يجري عن angular route guards
نحو "route guard" هي امكانية الوصول للroute
implement conditions في class الذي يمتلك
ال interface

ال guard هو built-in interface من خلاة نقر بخط
عن "route accessibility" في route configuration

انواع ال guard

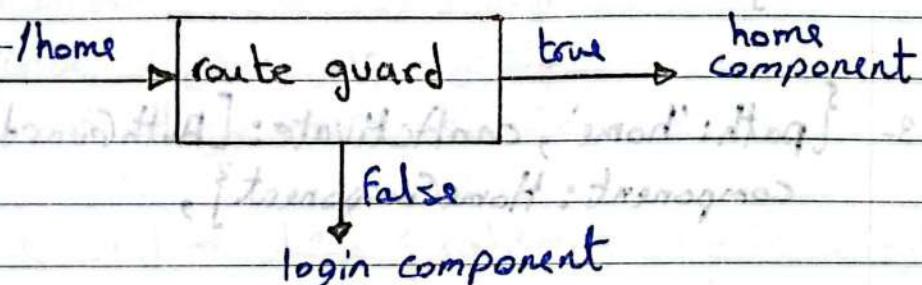
CanActivate

CanDeactivate

CanLoad

CanActivateChild

Resolve



return boolean function guard
false أو true

أغسطس

٢٨ أبيب ١٤٣٦ هـ

١٩ شوال ١٤٣٦ هـ

Example

٨- ١- `npm >> ng g g auth`

٩- ٢- `auth.guard.ts`

١٠- `export class AuthGuard implements CanActivate {`

`- constructor(private a: AuthService,
private r: Router) {}`

`canActivate(): boolean {`

`if (this.a.userData.getValue()) {`

`return true;`

`} else {`

`this.r.navigateByUrl('/login');`

`return false;`

`}`

`}`

٤- ٣- `{path: 'home', canActivate: [AuthGuard],`

`component: HomeComponent},`

٥-

٦-

August

Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon
١	٢	٣	٤	٥	٦	٧	٨	٩	١٠	١١	١٢	١٣	١٤	١٥	١٦	١٧

variables أو observable
methods أو observable

الاثنين

Monday

٣

٢٠١٥

١٨ شوال ١٤٣٦ھ

٢٧ أبيب ١٩٢١ق

أغسطس

④ BehaviorSubject

ما هو ال subject

ال observer و observable subject في نفس الوقت، يمكنه نشر رسالة subscribe و مستمر بـ notification data كـ object نسخة

الفرق بين ال observable subject وال subject ، بين casting و multi casting

عند subscribe كل بياضه رسالة uni casting خاصه بـ observable

عند execution subject يعين كل subscribers ما يـ listen بالترتيبه بتتابعه يعن كل subscriber يستقبل عالم observable بتبع الـ訂閱

لـ observable subject مثل حول الـ multi casting من uni

ما هو ال BehaviorSubject

هو نوع من انواع ال subject ، لـ ميزتين

① بـ بـ initial value مـ حـ اـ مـ

② بـ بـ value اـ حـ دـ اـ مـ

السبت	الأحد	الاثنين	الثلاثاء	الأربعاء	الخميس	الجمعة	السبت	الأحد	الاثنين	الثلاثاء	الأربعاء	الخميس	الجمعة	السبت	الأحد	الاثنين
٣١	٣٠	٢٩	٢٨	٢٧	٢٦	٢٥	٢٤	٢٣	٢٢	٢١	٢٠	١٩	١٨	١٧	١٦	١٥

August

CS

CamScanner

المسحوبة ضوئياً



2

الأحد

Sunday

2015

أغسطس

٢٦ أيلول ١٤٣٦ هـ

١٧ شوال ١٤٣٦ هـ

7

محوط

8 **BehaviorSubject** إلّا من نوع **property**
 9 يمكن ايجي الـ **value** بتأثرة من خارج **next()**
 10 وأقر اخر **value** فيه من خارج **next()** فقط.

10 **ex:** `userData = new BehaviorSubject(null);`

11 ~~`userData = 20;`~~ ✗

12 `userData = next(20);` ✓

1 ~~`console.log(userData);`~~ ✗

2 `console.log(userData.getValue());` ✓

3 **subscribe** يُخبر بآي تغيير للنرم

4 `userData.subscribe(() => {`

`if (userData.getValue() ==`

5 `} else {`

`}`

6 `}`

7 `});`

8 `});`

9 `});`

August

Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

CS

CamScanner

المسوحة ضوئياً

⑤ Route parameter

`routerLink = "/details"` ← المدارك route الـ "details"

`[routerLink] = ["/details", p1, ...]` ← parameters + route الـ "details"

(ex) (1) `{path: "details/:x/:y", ...}`

(2) `<div *ngFor="let m of movies" >`

`<a [routerLink] = ['/details', m.id, m.name] >`

`m.id` will be `url` of `the movie` with `id` `m.id`

`m.name` will be the name of the movie

``

`</div>`

(3) import {ActivatedRoute} from '@angular/router'

constructor(private ar: ActivatedRoute) {}

currentId: number = 0;

currentName: string = '';

itemDetail: any = {};

ngOnInit(): void {

this.currentId = this.ar.snapshot.params.x;

this.currentName = this.ar.snapshot.params.y;

route is a class from ActivatedRoute which has parameters in its constructor.
it receives url snapshot active route object from Angular Router.
it has methods to get parameters from url,
such as params, queryParams, queryParams, root, url.

٧ التجربة

```

constructor(private _ar: ActivatedRoute){}

٨ x: any = this._ar.snapshot.params;
٩ ngOnActivate(): void {
  console.log(this._ar.snapshot.params);
  console.log(this._ar.snapshot.url);
}
  
```

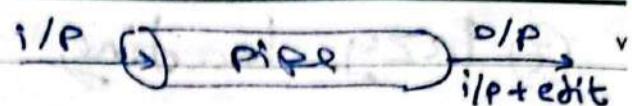
١٠ current URL من ال snapshot يساوي ActivatedRoute
 ١١ يتطلب object في browser engine URL يتكون من ال parameters
 ١٢ object property لـ params

٢ (ex) www.example.com?x=1 & y=50

٣ console.log(this._ar.snapshot.params);
 ٤ { x: 1,
 y: 50 }
 ٥ { f: example }

٦

⑥ search pipe



الـ input عبارة عن function pipe ويرجع
بعد تحريل معين على output

1-) `ng g p search`

2) `search.pipe.ts`

`.transform(movielist: any[], term: string) any[] { }`

return `movielist.filter(item => item.title.includes(term))`

3) `<input [ngModel] = "term" type="search" >`

`<div *ngFor = "let m of movielist | search: term" >`

`</div`

4) `term: string = ''`

`return movielist.filter(item => {
return item.title.toLowerCase().includes(term.toLowerCase())
})`

يوليه

٢٢ أبيب ١٤٣٦ هـ

١٣ شوال ١٤٣٦ هـ

٧) lazy loading

- ١- كل ممكن نسته ، أكثر من module او ينفع .
 - ٢- امر دوال module هو عبارة عن set of components
 - ٣- ازاي ندخل module بارتخاف ال CLI
- » ng g m setting --routing ←
• module اسم المودول

- ٤- ازاي ندخل module . بعض component
- » ng g c setting / profile ←
• اهم الملفات المدخلة

- ٥- cli يدخل components بال module لاعز ال ①
- » ng g m setting --routing ←
» ng g c setting / profile ←
» ng g c setting / password ←

- ٦- module يدخل routing → routing path ②
- app-routing.module.ts ←
• من هنا الـ router يتابع الـ path

{ path: 'setting' , loadChildren: () => import('./setting/setting.module') }
promise يرجع الى import method ←

```
... { path: 'setting' , loadChildren: () => import('./setting/setting.module') .then (r) => r.settingModule }  
(result)
```

July

Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

{path, 'setting', loadChildren, ...} .
عندما ندخل setting module للـ load .
الـ (setting) في الكـ routing .

. lazy loading ← اسـ تـ .
يعنى منـ هـ lazy loading .
. component module يستـ غـ lazy loading .

① ng g c c1
② ng g c m1/c1

" الاول يـ سـ يـ " . component generate الـ lazy loading .
. module m1 .

الـ app module هو الـ root module .
وبـ اـ الـ modules يتـ هـ منـ هـ .
. child module .

② يـ عـ اـ ؟ lazy loading concept .

request منـ هـ module للـ loading .

ازـ اـ عنـ طـ بـ اـ اـ اـ نـ عـ لـ اـ اـ مـ i~ p~ o~ r~ t~ .

- loadChildren property .
. imports promise method .

the result of dragging and dropping
multiple components into the root component
will result in the following .



٧) اهم نتائج lazy loading (Q)

٨) لوار scale يتابع ال project بعكس آليه ،
وأحياناً نقسم ال project لذكز من module .

٩) لية نستخدم ال lazy loading (Q)

١٠) تأثيره على performance (Q)
١١) module loading مثل lazy loading الـ معين إلـ ما نحتاجه / request مـ

١٢) module معناها هو lazy loading concept .
١٣) اطلبـة / لورـة .

١٤) SPA بـيـعـلـ loading كلـ الـ modules تـعاـديـاـ
أولـ الـ application يقومـ بالـ task معـ اـ لـ اـ رـاعـ الـ كـبـيرـ
الـ الـ فـيـرـ رـاوـتـ لـ تـغـيرـ ، الـ الـ loading time ، بالـ task صـنـاعـ الـ وـيـاـرـ
الـ الـ lazy loading دـيـنـيـلـ ، بالـ task صـنـاعـ الـ وـيـاـرـ performance
يـعـلـ اـمـلـ الـ load الـ اـ حـامـيـةـ دـيـنـيـلـ .

١٥) اـنـاـ نـعـلـ الـ module الـ loading الـ نـسـاجـتـ (Q)

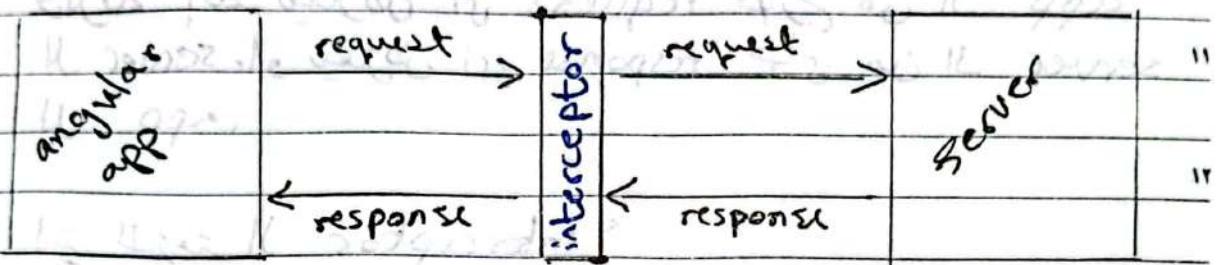
١٦) عن طـريقـ الـ dynamic import الـ خـالـلـ انـ اـمـنـاـ
تـطلـ الـ routing مـعـ الـ loadChildren
الـ الـ route's property هو loadChildren بالـ عـاصـ ، يـاخـدـ
الـ الـ import method يـرجعـ فـيـهـ الـ arrow function
الـ الـ module class يـسـطـرـ الـ module .



⑧ interceptor .

اعتراض / المترمن

الـ client side interceptor هو مbari عن layer بين الـ request من server side ، يعني بيعتزم طريق الـ request والا لـ angular app . لـ server . لـ request قبل ما يتبعد للـ server .



لـ angular app

⇒ neg g interceptor apiKey

* الـ http request كل الـ layer هو interceptor الا server . طالع من الـ angular app قبل ما يتبعد للـ server .

* عن طريق الـ interceptor نعمل تتعديل على كل الـ request قبل ما يتبعد للـ server .

* الـ interceptor ميقـر ، يعدل على request مباشرـاً ، لكن بيأخذ clone من الـ request ويتعديل عليه .

* لـ الـ interceptor لازم نربطـه بالـ app من app module .

٥٠

25

السبت

Saturday

2015

يوليه

١٨ أبيب ١٤٣٦ هـ

١٧ شوال ١٤٣٦ هـ

٧) (٢) دو ال Interceptor

- layer concept هو Interceptor يندرج من Interceptor على server side وال client side

٩) (٢) اجر فکرة ال Interceptor

ذكريتة انه ي Intercept اي request خارج عن ال "server" او ي Intercept اي response جاء من ال "server" لل app

١٢) (٢) اجر لازمة ال Interceptor

من خلال ال Interceptor نقرر فعل اي تعديل على اي request طالع من app باتجاهنا، قبل ما يصل الى server

- نقرر نضبti اي request api key لاس طالع
- نقرر نحط اي token في اي request طالع
- نقرر نعدل اي صاريف من request من header
- نقرر نعدل اي صاريف من ال header

٦) (٢) اجر دو ال intercept() function
هو دو ال Interceptors / layer ، بياخذ حاجتين :
١) ال request : سواء الـ طالع او الـ جاكي .
٢) التغيل الـ next : دو التغيل الـ next عمل حفـر الـ next

July

Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

CS

CamScanner

١- using a interceptor apikey

٢- apikey.interceptor.ts

```
import { Injectable } from '@angular/core';
import { HttpRequest, HttpHandler, HttpEvent } from '@angular/common/http';
import { Observable } from 'rxjs';

@Injectable()
export class ApikeyInterceptor implements HttpInterceptor {
  constructor() { }

  intercept(request: HttpRequest<any>, next: HttpHandler): Observable<HttpEvent<any>> {
    let newRequest = request.clone({
      url: request.url.replace('key=', 'key=123456')
    });
    return next.handle(newRequest);
  }
}
```

٣- app.module.ts

providers: [

```
{ provide: HTTP_INTERCEPTORS,
  useClass: interceptor,
  multi: true
},
```

الآن نفتح المتصفح ونذهب إلى الموقع
https://httpbin.org/headers
نلاحظ أن المتصفح قد أضاف مفتاح key=123456
إلى رأس الطلب (header) وذلك لأن المتصفح
يعمل كـ proxy (متوسط) بينك وبين الموقع.

(٦) production

٣ خطوات

- اول خطوة:** نزوح ملحوظ الى `index.html` ،
`<base href="/" />` ، منعيف قبر ال slash من الا "/" .
- ثانية:** نطبق او `relative paths` .

(٧) `<base href=". /" >`

- ثانية خطوة:** نزوح ال الـ `app routing` الى `useHash` او `app-routing.module.ts` .
- ثالثة خطوة:** `# hash` قبل كل `route path` .

(٨) `@NgModule({})`

```
import { RouterModule.forRoot (routes, {useHash: true}) },
```

3 = 4

5 6

7 8

- ثالثة خطوة:** استخدم الـ `webpack` و دى سهل من الـ `cli` .
- fourth step:** `configuration` اى اضافة بتحويل / بتجهيز الـ `deployment` او `angular` .
- 1- تجفيف ملفات الـ `css` من ملف `index.html` .
- 2- تحويل الـ `ts` الى `js` .
- 3- تحويل الـ `css` الى `sass` او `less` .

لو فرق error في المشروع

(٩) `ng build`

امرا الـ `build` من هي تفعل .



ما هو ظاهر

١) أمر ال build متى يستعمل لو خطاً اي .
من امره تردد .

٢) ناتج أمر ال build انه تكون folder جديد اسمه
(dist) يضم كل files distribution ، و هو هو ال
الكل هنعملة deployment .

١١

١٢

١

٢

٣

٤

٥

٦

٧

٨

٩

July

٢١

21

الثلاثاء

Tuesday

2015

بولييه

١٤٣٦ هـ

١٢ أكتوبر ١٩٢١ ق

٥ شوال ١٤٣٦ هـ

- Angular
- Material
- Design

Session 8

1. introduction
2. getting started
3. generic material module
4. Typography
5. Components

12

1

2

3

4

5

6

July

Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	٢٠

CS

CamScanner

المسوحة ضوئياً بـ

① Introduction

ما هو Angular Material؟
هي مبادرة من UI Components library متاحة جاهزة
، مبرأة اذ اقتراحته مشروع بـ تسلسل سريعة.

الآن نظرنا على library بالـ angular material
لقد ادركنا ان الـ Angular Material مسؤول عن خصوصيات

② Getting Started

لابد من نضري على angular Material من المشروع

⇒ ng add @angular/material

١- انتي الى الـ theme الذي انتي عليه؟

٢- هل تحتاج الى typography

٣- هل تحتاج الى browser animation

app.module.ts

angular.json

style.scss

package.json

index.html

٤- انتي الى material component في ٣ خطوات

١- ادخل material component module الى app module imports

٢- اكتب اسم الـ module في imports array

٣- اكتب الـ html code في المكان الذي احنا عايزين نظهره.

يوليه

١٢ أبيب ١٤٣٦

٥ شوال ١٤٣٦

③ generic material module

يُدخل في المódule الكروة مثل Import in module فتح كل material modules لكي نقدر من تخصيصه

app module

① >> ng g m material --routing

② add exports array in @NgModule

(ex)

const MaterialComponents = []
 @NgModule({})

imports: [] module ||
 exports: [] component ||
 all the components

MaterialComponents
]) ||

③ import material module in app module.

ابدأ العمل من كودك لحرف تنظيم من أكثر

جتنى تخلى المódule يكون

يعنى نقدر نكتب فيه generic

الـ components

6. جرب الموديل في الموديلات

لذلك، أدرج الموديلات في الموديلات

2. ثم أدرج الموديلات في الموديلات

3. ثم أدرج الموديلات في الموديلات

July

Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

① Typography

ما هي إلا تبع CSS classes material design . i.e p, h6 ... h2, h1 ... text

ex) <p class="mat-display-1" > - </p>

<p class="mat-headline" > - </p>

<p class="mat-title" > - </p>

mat-body

mat-caption

(ماتروطة) ١- داعياً ببداية بكلمة (mat-)

٢- الـ classes دول مكان تعلمها باديلك واصحه

بالـ font-size ، لكن درجة بيور على وقتنا ☺

⑤ Components

أهم الـ components

angular material من

Button

Button toggle

Icon

Badge

process spinner

navbar

sidenav

menu

Dialog

List

gridlist

expansion panel

card

tabs

stepper

input

select

basic autocomplete

filter autocomplete

checkbox, radio box

datepicker

Tooltip

snackbar

يوليه

١٠ أبيب ١٤٣٦ هـ

١٧٢١ ق

أيشوال ١٤٣٦ هـ

7

ملحوظة

① اى حاصل تبع material design

② مطوابق ثابت التعامل مع اى component

③ اعملت app module او import من ماجستير

او اعملت module او import من ركوب

او import as module وائل ال generic من ال

app module

④ اخيرة من ال .html

2 that's it

3 what's the platform engine

4 delegation class

delegation method

5 delegation library

6 delegation and base

delegation

7 delegation

8 delegation

9 delegation

10 delegation

July

Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	25	26	27

أية الـ مسحاج متذكرة بعد كل يوم
أية الجديدة الـ يغتصل تتلاطه

١- Angular Material Design

٢- Angular primeNG

٣- RxJS (reactive programming)

٤- NgRx (state management)

٥- Unit Testing

٦- Angular + Firebase

٢٢ رمضان ١٤٣٧

٢٢ يونيو ١٩٣٨

July 2016 ٢٠١٦ يوليه

Friday الجمعة

الجمعة

١

PKJS

-١- ايجي RXJS

Reactive Extensions for JS
هي امتداد لـ asynchronous code library التي التعامل مع الـ RXJS يتحقق ذلك من خلال خوارزمياتها.

-٢- ايجي المزدوج الارسالي من الـ RXJS

التعامل مع الـ asynchronous code بـ شكل ابسط
من خلال الـ RXJS Tools سبقوها.

-٣- دليل الـ rxjs وـ es6

الـ rxjs مكتبة من الـ ٨٨ الـ API يتبع الـ ECMAScript 6
هي synchronous code ولكن الـ ٦٤ features موجودة في الـ browser
متى تدخل asynchronous code

-٤- سيناريوهات

(١) execute task / code
يعني تكرر من الـ task / code
تتغير الـ arguments من بعض.

(٢) objects من الـ ٤٠ نوع معين من الـ function

Name _____

Tel. _____

Email _____

يونيه ٢٠١٦

٢٥	١٨	١١	٤	السبت
٢٦	١٩	١٢	٥	الأحد
٢٧	٢٠	١٢	٦	الاثنين
٢٨	٢١	١٢	٧	الثلاثاء
٢٩	٢٢	١٥	٨	الأربعاء
٣٠	٢٣	١٦	٩	الخميس
٢٤	١٧	١٠	٣	الجمعة

أحد اهم مكونات RxJS هي observable

أولاًً "أهم" من observable هو observable subscription observer

• من data generator هو observable

- يمكنه subscribe على data emit

• values generate function أو يمكنه subscribe بنقل

• observable run في function subscribe

- values في emit في observable
- data / values في generate

• ومن خلال observable يمكنه subscribe method

- emit أو generate على observable

- subscribe على observable
- run على observable
- كل ما نعمل في observable هي asynchronous task
- API التي لها emit يمكن منها نظرية

properties

- object قبل subscribe على observable
- observable وكل مساحة يمكنها من observable
- observableObj.subscribe({next: ..., emit: ..., error: ...})

3);

↳ notifications

يونيه ٢٠١٦

June

٢٠١٦

الثلاثاء

Tuesday

٢٣ رمضان ١٤٣٧

٢١ يونيو ١٣٩٢

٢٨
28

١- افاهم ايجز عن ال observable وال subscribe

٩- ال observable هي observable function هي دالة function التي تتبعها logics ، الفرق بينها وبين ال function العادي انه تصدر فعل emit نسخ من ال data مع مرور الوقت.

١٠- ال observable لوصفا مبتداً من فعل صاصه ، للزمن فعل subscribe على سبيل المثال .

١١- منصر نتعامل مع ال data الـ Observable من ال observable او من ال errors الـ Observable من خلال او observer object .

الشرح الـ ١ Quick start عن الـ RxJS
الشرح الـ ٢ جاري تفاصيل آخر عن !

- ① observables
- ② subscriptions
- ③ observers
- ④ creation functions
- ⑤ pipeable operators
- ⑥ subjects

Name _____

Tel. _____

Email _____

يونيه ٢٠١٦

٢٥	١٨	١١	٤	السبت
٢٦	١٩	١٢	٥	الأحد
٢٧	٢٠	١٣	٦	الاثنين
٢٨	٢١	١٤	٧	الثلاثاء
٢٩	٢٢	١٥	٨	الأربعاء
٣٠	٢٣	١٦	٩	الخميس
٣١	٢٤	١٧	١٠	الجمعة



① Observables

١- ابرهاد ال stream

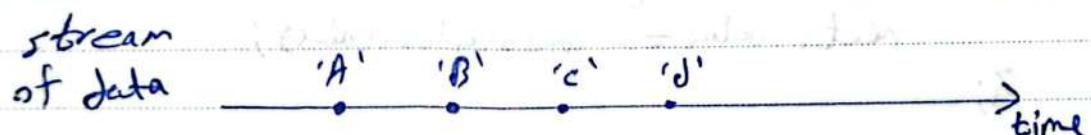
ال observable ينتهي بـ stream ،
ال observable يقر بـ emit أي ان من ال values
يُنطَّلِقُ منها مختلفاً .

ال stream هو سلسلة من ال data يظهرها
عن نقطه زمنية مختلفة مع مرور الزمن ، ونفترض access
ال data أو أن react على ميل من الوقت / الزمن الـ اك يتضمن
فترة فقط ، يعني لو الوقت ليس وظيفة data بعد كل
مرة منقارنة access غير ال data الـ اك خارج من الوقت
الـ اك بعد access قرابة .

٢- ابرهاد ال array

ال array هي مجموعة data موجهة طول الوقت ، يعني
نقدر نخالهم / نعمل access لأن data موجود في أي وقت .
انما ،

ال stream هي مجموعة data يتضمن ازمنة مختلفة ، ومنقارنة
نقدر على الـ data الـ اك يتضمن في الوقت الـ اك بعد access
فترة .



٣- ابرهاد ال stream المترافق

ال reactive programming هو ابرهاد ال approach
وهي الطريقة الـ اك ال observable يغير عليه .

٢٦
26

١٤٣٧ رمضان ٢١

١٧٢٢ بیونیہ ۱۹

June

Sunday

2016 ٢٠١٦

بیونیہ

الأحد

٤- ریlation اے observable بال stream

مے اے stream بیجل ونگت نکھت میں data میں emitting ہے اے observable کے زمانی مختلف ونگت و رکھر رہے میں بتھڑے.

میں data میں emitting ہے اے observable کے زمانی مختلف ونگت data میں reacts ہے میں بتھڑے یعنی لازم نہ تھا ال data بتھڑے میں رکھر استعمال معاضا.

٥- اے observable

با سچھ اے new Observable() constructor میں observable

نیست قبل چو () اے observable کے callback function

```
(ex) const obs1 = new Observable (subscriber=>{
    subscriber.next('A');
    subscriber.next('B');
});
```

اے observable کے

property کے properties میں object ہے کہ observable میں notification میں اے notification سختم پریغڈل ہے.

```
(ex) const observer = {
```

```
    next: value=> console.log(`Value: ${value}`);
};
```

Name

Tel.

Email

بیونیہ ٢٠١٦

٢٥	١٨	١١	٤	السبت
٢٦	١٩	١٢	٥	الأحد
٢٧	٢٠	١٣	٦	الاثنين
٢٨	٢١	١٤	٧	الثلاثاء
٢٩	٢٢	١٥	٨	الأربعاء
٣٠	٢٣	١٦	٩	الخميس
٣١	٢٤	١٧	١٠	الجمعة



CamScanner

المسحوة ضوئية

١) ايجـ الـ الفـ بـ الـ observable

الـ logic لـ observable هو مـجرـ object يـحتـوى عـى مـعـينـ وـيقـرـ بـ مـعـىـ data ، كلـ data emit بـطـبـقـ علىـ notification

- ٣ diff notifications يـقـرـ بـ observable
- next notification ①
- error notification ②
- complete ~ ③

reaction لـ observable يـوصـفـ الـ object

الـ exit علىـ الـ data الـ اـعـاـدـ

الـ execute فيـ اـقاـمـ الـ alـ بـ subscription

callback function run بـ اـخـارـ الـ observable

الـ observer جـداـ الـ observable وبـشـ تـقـيلـ الـ

جـداـ .

٢) الـ اـنـاـرـ نـجـلـ subscription

احـ observable منـ خـارـ الـ method

subscribe()

(ex observable.subscribe(observer);

٣) اـسـتـرجـ الـ process

subscription observable والـ دـخـلـ subscribe ①

observer observable الـ execute

٤) كلـ مرـهـ الـ next notification منـ خـارـ

الـ observable run بـ next callback

منـ خـارـ الـ observer وـالـ عـلـيـتـينـ دـولـ هـيـ تـحـرواـ

حـولـ الـ subscription

٢٤
24

١٩ رمضان ١٤٣٧

١٧ جلوونه ١٣٣٨

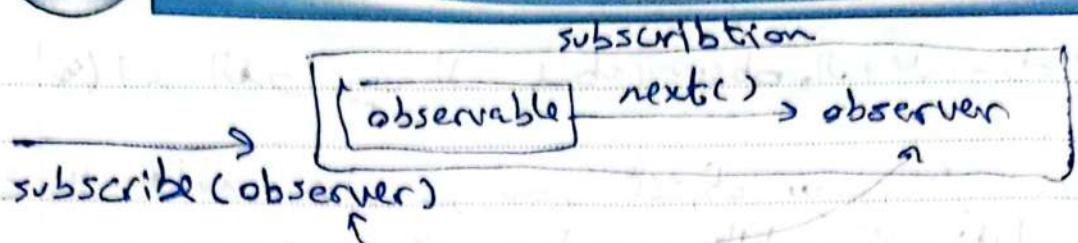
June

Friday

2016 ٢٠١٦

يوليه

الجمعة



أنت أو subscription بسيفخ (Q)

من خلال طريقيتين ،
• unsubscribe() method (١) باستخدام
او مستغفل سهل طفاف من خلال الـ logic (٢)
جهاز او observable منتهى عن طريق ارسال
complete notification او error notification

(Q) لمحظكم تناهى اى الـ subscription
memory leak او side effect اى ماحصلت

راجع الخطوات الـ ٥

- ١ new Observable جهاز باستخدام observable
- ٢ next notification object يرسل الى observer
- ٣ متصل observable الى subscription
- ٤ متصل الى subscribe method
- ٥ ستعمل الـ unsubscribe باستخدام الـ subscription

Name

Tel.

Email

يوليه ٢٠١٦

٢٥	١٨	١١	٤	السبت
٢٦	١٩	١٢	٥	الأحد
٢٧	٢٠	١٣	٦	الاثنين
٢٨	٢١	١٤	٧	الثلاثاء
٢٩	٢٢	١٥	٨	الأربعاء
٣٠	٢٣	١٦	٩	الخميس
٣١	٢٤	١٧	١٠	الجمعة



Q) از ار نسل میکار data را emit کن

با استفاده از $\text{next}()$ method در هر Observable میتوان subscribers بین تغییرات object سپرید. از observable برای بنیانگذاری callback استفاده شود.

```
(*) const observable = new Observable()
  subscriber => {
    ;
```

اگر logic را در subscriber میکنیم، observable میتواند subscriber را next کند.

Q) از اکسپریس Observable که میتواند logic را اجرا کرده و آنرا next کند.

execute خواهد شد subscription که observable را از logic برای subscribers منفصل کرده باشد.

برای subscribing observable از callback function استفاده شود.

each subscription is a separate independent observable's execution.



١٧ رمضان ١٤٣٧

١٥ يونيو ٢٠١٦

June

Wednesday

٢٠١٦

يوبيله

الأربعاء

١) انتقال الغرق بين الـ { complete , error , next }

exit data فتح انتقال data في next notification ①
 من وفور الـ observable complete ~ ②
 حامض ثانية تستبدلها ١.

٢) ميل فتحية حامض خطأ مفتوحة بالاتجاه
 من وفور الـ subscription ٢.
 دفع unsubscribe ١٢

٣) لفتح الغرق بين الـ { complete , error }

الاشتراك ينطوي على observable
 الاشتراك ينطوي على subscription ٢.
 ولكن الفرق إن الـ error يقترب بـ fail
 معانٍ ، وـ complete مماثل ٣.

٤) next ← يستحصل على داتا .
 values ← تتحقق بعد emit لـ values الـ notification .

٥) error ← يستحقنا لـ error أو fail .
 payload ← تقدر تغير payload ٦.
 events ← يستحصل على واحد بـ .
 subscription ← يستوقف عليه الـ ٧.

Name _____ Tel. _____ Email _____

يوبيله ٢٠١٦

٢٥	١٨	١١	٤	السبت
٢٦	١٩	١٢	٥	الأحد
٢٧	٢٠	١٣	٦	الاثنين
٢٨	٢١	١٤	٧	الثلاثاء
٢٩	٢٢	١٥	٨	الأربعاء
٣٠	٢٣	١٦	٩	الخميس
٢٤	١٧	١٠	٢	الجمعة



CamScanner

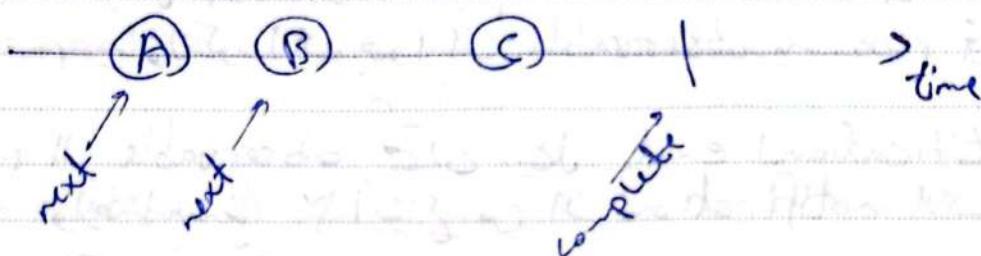
المسوحة ضوئيا بـ

-marbles diagram (Q. ١٧)

هي مفهوم أو يمثل حلول الـ observable والسيناريوهات المختلفة لـ subscription.

مكوناً للـ marble diagram

- A. next notification ①
- X error ~ ~ ~ ②
- | complete ~ ~ ~ ③
- time line & observable لـ ~ ④



يونيه

الاثنين

٢٠١٦

June

Monday

١٥ رمضان

١٤٣٧ هـ

١٣ بلوونه

٢٠



② Subscriptions

٩) اشرح رورة صياغة الـ subscription

١٠) اولاً، يمكننا معرفة observable، والمتغيرات observable هي run بصفتها، هو ببساطة جهاز logic متغيرين حركة.

١١) ثانياً، يمكننا observable خارج subscribe و هو دليل observable للكور (كـ جهاز execute).

١٢) ثالثاً، لا يتعلّق subscribe بمقدار (ستراتيفي)، بل هو observable run في observable.

١٣) رابعاً، الـ observable هو الذي يُعمل على observable (أو الكور) الذي هو من نوع notifications.

١٤) كل observable يُرسل next notification، العلامة التي تلي قيمة subscribe للـ observable's handler هي المكان الذي يتم إصدار next notifications.

١٥) observable's code لا next notification قبل running.

Name _____

Tel. _____

Email _____

يونيه ٢٠١٦

٢٥	١٨	١١	٤	السبت
٢٦	١٩	١٢	٥	الأحد
٢٧	٢٠	١٣	٦	الاثنين
٢٨	٢١	١٤	٧	الثلاثاء
٢٩	٢٢	١٥	٨	الأربعاء
٣٠	٢٣	١٦	٩	الخميس
٣١	٢٤	١٧	١٠	الجمعة



CamScanner

المسوحة ضوئياً بـ

١) لو اد observable دعوب observer if error handler will invoke دعيم على عل observable execution بعد كده دعيم اعلاف الـ subscription يعنى على دعيم execution ، بعد كده observable running begins . observable will cleanup ورقة دعيل Teardown

٢) لو اد observable دعوب error دعيم عن اد step دعيل

٣) حاصل اد observable دعوب error دعيم عن اد observable subscription دعوق الـ observable كده فالـ unsubscribe() دعيل الـ teardown

٤) فيـةٌ Subscription طريقة دعوق بـيل اد

- error notification دعوب ١
- complete دعوب ٢
- unsubscribe() method دعوب ٣

٥) اد دوالـ و ازاي نعملـ

ـ هو مجرد كورـ / logic بـينـ دعـات دعـيلـ
ـ مثلـ `settimeouts()` دعـوقـ `cleanup()` دـعـيلـ
ـ او دـعـقـ `data` دـعـيلـ `memory` دـعـيلـ

ـ return دـعـيلـ function دـعـيلـ `teardown` دـعـيلـ
ـ observable دـعـيلـ `callback` function

ـ `→ observable's callback returns teardown logic/function.`

يوليو ٢٠١٦

June

١٣ رمضان ١٤٣٧

١٨
18

١١ يونيو ١٩٣٢

Saturday

السبت

③ Types of Observables ١

• عدنا نوعين من ار وها
 • hot observable
 • cold observable

• cold observable
 • hot observable
 • واجه املاك الفرق بينهم

HOT

COLD

كل "hot" يعني كل observable ينبع منه مطالبات subscribers قبل execution من قبل source.

لـ events كل subscriber يستقبل نفس stream of values.

جديدة خاصة بها ومنفصلة عن باقى ار subscriber.

APIs مثل Ajax

Name _____

Tel. _____

Email _____

يوليو ٢٠١٦

٢٥	١٨	١١	٤
٢٦	١٩	١٢	٥
٢٧	٢٠	١٣	٦
٢٨	٢١	١٤	٧
٢٩	٢٢	١٥	٨
٣٠	٢٣	١٦	٩
٣١	٢٤	١٧	١٠

السبت
 الأحد
 الاثنين
 الثلاثاء
 الأربعاء
 الخميس
 الجمعة



CamScanner
 الممسوحة ضوئياً بـ

١٤٣٧ رمضان ١٢

June 2016 ٢٠١٦ يونيو

١٧٢٢ بُـؤـونـه ١٠

Friday

يـوـنـيهـ الجمعة

١٧

*Hot
events*

(A) (B) (C) (D) →

٥

(A) (B) (C)

١٨

time

(B) (C) (D) →

٦

(A) (B) X

٩

(D) →

٧

(A) (B) ١٠

١١

كل الـ subscribers يستقبلوا
stream of values (تسلسـل الـ values)
نفس الـ source ، وستـقبلـها
نفس الـ values (نفس الـ time).
الـ الوقت هو وما يـمثلـها
ـcreateـ يستـدلـ منـفصلـ .

كل الـ subscribers يستقبلـها
values (تـسلـسلـ الـ values)
ـخـاصـةـ بيـتهـ .

(cold)

الفرقـ الـ caseـ مـابـينـ الـ cold / Hot observables
ـخـواـصـ الـ dataـ الـ coldـ بـيعـبـدـ مـنـفصلـ كلـ subscriberـ . إنـهاـ الـ hotـ بـيعـبـدـ مـنـفصلـ كلـ subscriberـ . إنـهاـ الـ coldـ بـيعـبـدـ كلـ الـ streamـ of~ dataـ وـ تـغـيرـ الـ الوقتـ .

ـ لـ وـ دـنـعـلـ emittـ منـ داخلـ الـ unobservablesـ .

ـ hot ~ ~ ~

١٦
16

١١ رمضان ١٤٣٧

June

٢٠١٦ ٢٠١٦

يونيه

٩ يونيو ١٤٣٨

Thursday

الخميس

④

Subject

لِهِ سُوال subject

دوّن observable، نوع خاص من الـ observable values لـ multicasting . subscriber ، observer من

هي → الـ event emitter ، وجهة النية التي كل طابعات emit يقوم بذلك event لـ listeners الى ملحوظ

الـ observer هو subject و observable متى الوقت ، بالتأكيد ① تقدّم subject مارك observes ، الذي ما يحصل مع

② تقرر بـ next() ، errors() ، وال () complete() مع اـ observer ، الذي ما يحصل مع اـ observer العادي.

subject = observable + observers

Name

Tel.

Email

يونيه ٢٠١٦

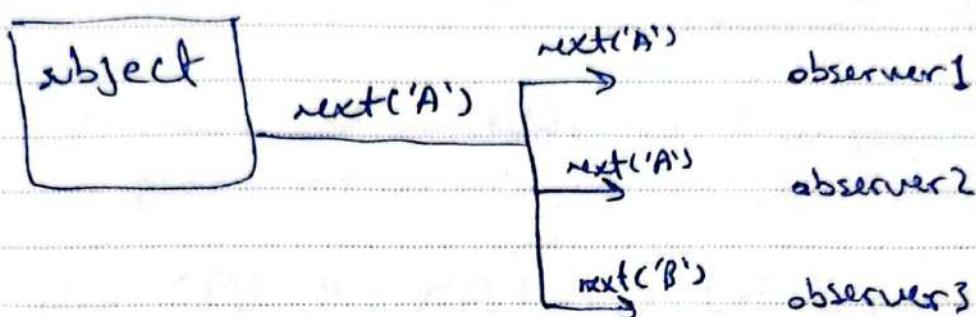
٢٥	١٨	١١	٤	السبت
٢٦	١٩	١٢	٥	الأحد
٢٧	٢٠	١٣	٦	الاثنين
٢٨	٢١	١٤	٧	الثلاثاء
٢٩	٢٢	١٥	٨	الأربعاء
٣٠	٢٣	١٦	٩	الخميس
٣١	٢٤	١٧	١٠	الجمعة



CamScanner
المسوحة ضوئياً

{ cold / hot observable دل ال subject دل observable

multicasting دل observable
يعنى بجعل notification data وال information available
متاح لـ كل ال subscribers الى علية ويسـتقبلوا
عنـ اـل data مـ نفس الـ time .



ابـ الـ cri بين الـ observable دلـ subject

الـ observable دلـ observable دلـ subject دلـ subject
وارـ observer يـعـنـ تـقـرـيـبـ عـلـيـهـ subـscribeـ بالـتاـكـ
نتـاجـ اـلـ notification اـلـ جـاـرـ مـنـهـ
ونـتـدـرـيـجـ تـحـمـيـلـ اـلـ (next)ـ مـعـ كـلـ مـيـاـنـسـ خـلـاـلـ
الـ multicastedـ subjectـ وـ الـ nextـ notificationـ مـيـخـرـ subjectـ
لـ كـلـ الـ subscribersـ .

ابـ الـ cri بين الـ behaviorـ subjectـ وـ الـ subjectـ

الـ statefullـ behaviorـ subjectـ هوـ سـتـخـرـ عنـ اـلـ
emitـ يـعـنـ بـيـخـرـ آـطـرـ اـعـلـيـهـ valueـ subjectـ
الـ subscribeـ لـ الـ behaviorـ subjectـ لـ الـ memoryـ عـلـيـهـ
لـ اـنـ اـعـلـمـ emitـ behaviorـ subjectـ behaviorـ subjectـ
لـ اـنـ اـعـلـمـ updateـ subscriberـ وـ صـيـغـهـ الـ الجـمـعـهـ
حيـثـ يـعـنـ مـتـاجـرـ آـنـ updateـ معـ باـمـ اـلـ