**Web1**

**SPA: single page application**

**Angular is a collection of js files makes your web development easy and smarter.**

🡪angular.js was totally built using js files & all our code was .js files

2015: Microsoft (typescript), google adopts typescript as first citizen programing language web

Angular 2: totally re-built (angularJs protect is not upgradable to Angular2 ) due to typescript.

Angular 4 🡪 Angular 6 🡪 Angular 8 🡪 Angular 9🡪 Angular10

**🡪Angular make the program very fast because it have only single page app**

**🡪angular recognize a new tag you were typing it**

**🡪reactJs one or two files js**

Global packages

Ng new file

Ng serve -o

Ng build –prot

Ng g c (generator component)

Npm I Nodemon -g

Npm I -g typescript

Npm I -g @angular/cli

1-Install this package



this is a package continue many other packages and one of these is package.json whish contain all independencies name for that if the connection is slow and the installing was stopping you can continue it by npm i

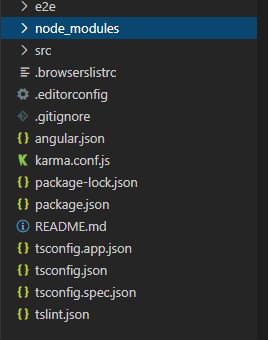


2 – create a new angular project

Ng 🡪 npm i

🡪download with github

Ng new go to “github” and download seed (بزرة) angular project with npm i



Tolled the webpack where it should be make in the index.html

Related with the e2e

If you want to learn more go to angular protect

End to end testing: you write a software testing your code

It is a file than write inside it what the file should not push it when you want push your projectct on github

It not essential whish you write information about your project

Contain rules whish Makes your code reading

(it Makes by google)

Depends on it t know for any version of es it should be compiler

Spa: single page application(one html file)

Angular 🡪 Electron app (visual studio code)- mobile app – web app.

Ng serve -o:

1

**Tsc**: it converts .ts files into .js files.

2

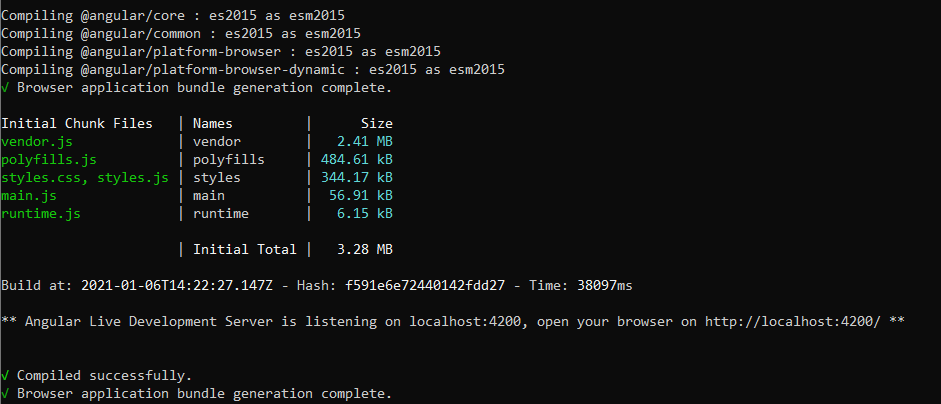
**webpack**: it packs all converted .js files

3

**express**: starts a development / local web server

4

**nodemon**: it does monitor of “src” folder

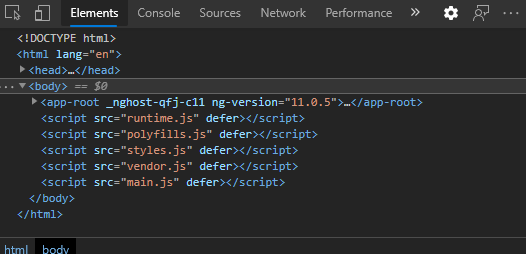


4

3

2

1

Enter the localhost 🡪 *Fn* + F12 🡪 

Our code

styling

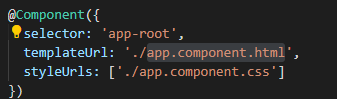
Power low when you open enternet explore

Contain all angular code which google makd

Webpack is who make this js file here.

|  |  |
| --- | --- |
| **Require** | **Import/export** |
| Come with all the file | Come with only the export file |

Class is a collection of properties and functions\

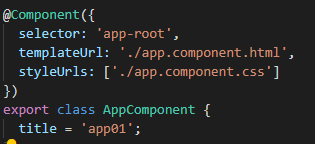


String: “ahmad”;

Number: 2;

Object: {selector: 'app-root',  templateUrl: './app.component.html', styleUrls: ['./app.component.css']}

Propertie array



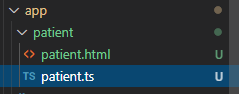
In programing world is a tag

Decoration: function contain an object

App root :you use a html file and make at in the element (app root) so when you called the element it called the html file

For create a new tag :

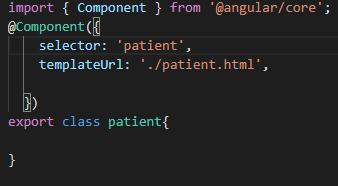
First: create a folder 🡪 create a .ts file 🡪 create a .html file



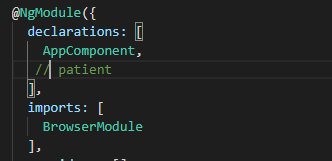
Second:folder/app/patient.ts 🡪 create decoration 🡪templateUrl: ‘.html file’



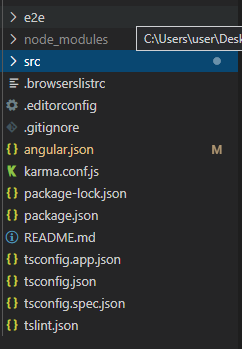
Go to the @angular/core and import the {component}

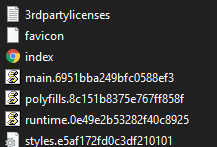


go to the **app module.ts** and declareted it.



When you type ng build –prod (build for app for production) it creates a file called it dist (distrbution) contain the files make it in the browser.



=

New tag2

New tag1

New tag

App root

Index.html

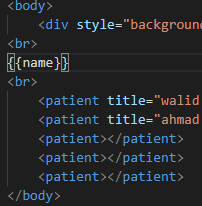
Runtime is an angular

= all independencies in the node module folder

When the browser see a new tag the runtime.js find the .html file whish it introduce in the component and replace the tag and make the contain of the .html file after go to the declaration. But before all these the run time create a new object by a class

//The constructor will directly fire once the runtime create an object from the class (instance)



When we make a property in the class

If before all these the run time create a new instance by a class by second so we can use it in the appcomponent

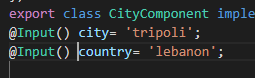
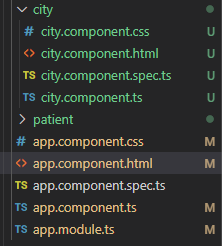
All steps in the first session will be replace by ng g c

Dad 🡪 son @input

Son 🡪 dad @output

**@input**

After create the component we can start to relate the component child with the parent;



Go to he .ts file and create a property has @input decoration

1



2

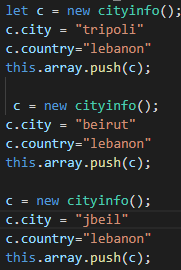
Create a new file .ts (for the data) and export a class

And the propertie is the data



3

Go to the app.component.ts and create an array



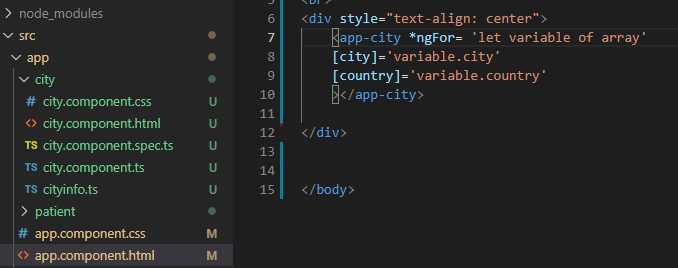
Create new variable and make the data

“city” “country”(step2) and push it

4

5

Go to the app.compponant.html and type:{\*ngfor= let variable of array}



اعد رسم الآريي بعدد الداتا الموجودة

ثم عرف الداتا

**@output**

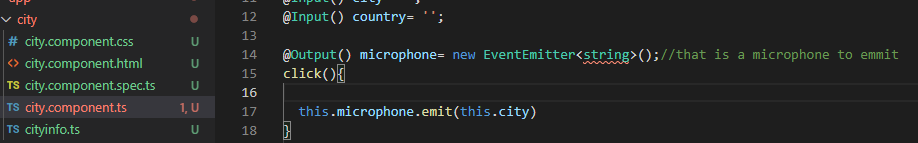
First go to the component’s .ts file and add an propertie



Make that propertie in a function to actived it

2

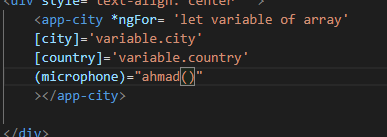
هون عيط الولد و الاب سمعوا بس ما رد



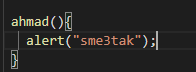
Go to theparent html to type the replay after create a function in the parent .html

3

هون الاب رد



App.componant.html



App.componant.ts

For the approot replay to the component child with what he say we will have to give a parameter for the function replay called ($event) in the .html and a parameter x to the same function in the .ts

And we can create a selective to replay what he said

>app.component.html 🡪{{selective}}

>app.component.ts 🡪 selective:’’(new property in the main class)

And in the function (replay) we type this.selective = x

**Service**

* the service is a class then any file can import it
* to create a new service (ng g s) but it should declare in the provides
* frequently it’s using to make inside it the array

If you have a class with a constructor, and this constructor has a parameter you don’t can create an instance with that class without you give it a parameter

Rule 1

Instance=result

Rule 2

We can create a new property to the class by give to the constructor a parameter (private propertyname: propertytype)

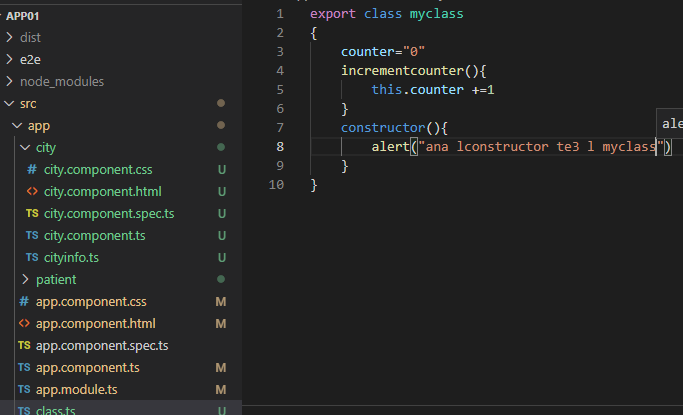
It’s not interesting to give the property type

 constructor(private title: any) { }

but we can’t create a instance (rule 1)(video >20210109>02:10)

runtime.js can’t create an object

for that we create a new file called class.ts and we export a class

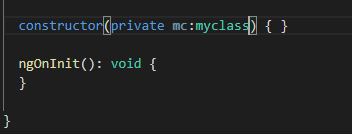


Then go to the city.componant.ts and add t a parameter to the constructor

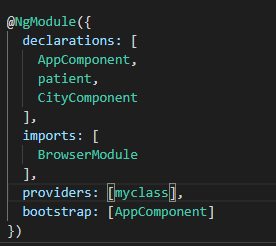
Dependency: private

Infection: mc:myclass

That equal to-> let mc=new myclass()



Next go the app.module.ts and declare my class in the provider



The main

In the providers the runtime create a instance for once time and that instance was intersection with all the componant

For that we tell the run time : before you run the app go and create an instance with the my class (proveders) and if you want to create a component need the parameter my class use the instance whish you created it.

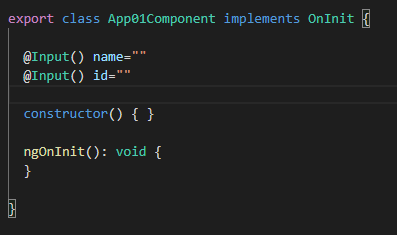
**Note:** this way is singleton 🡪 create object 1 time only and uses with all

@injectable decorator is a way to tell angular the dependency injection is possible with the class

**Note: angular contain a complete service to use it (example: Route / ActivateRoute)**

Implement

Implement interface is an way in oop to say it should contain all functions which its in the interface



Constructor > ngOnInit > instance

**How call this function?**

🡪runtime called it

ngOnInit access the property and the value but the constructor can’t

= the correct place to read input properties

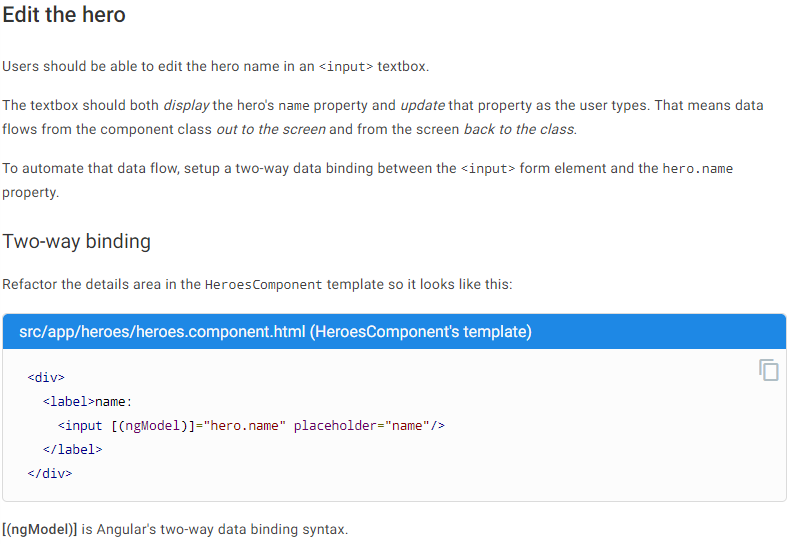
**For that the constructor is only for the services**

**Bootstrap**

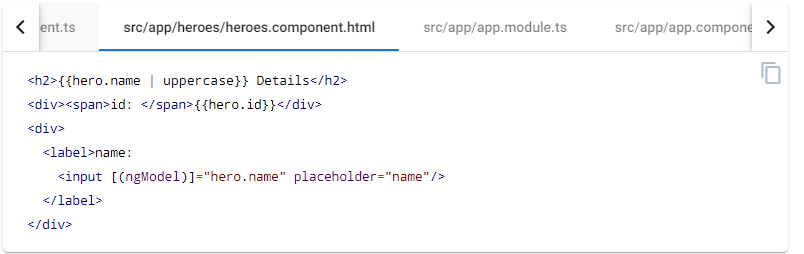
Npm I bootstrap and go to the bootstrap and copy anything you do.”node\_modules/bootsrap/dist/css/bootstrap.min.css🡪.min the same of .css but without white space

Summary

* You used the CLI to create a second HeroesComponent.
* You displayed the HeroesComponent by adding it to the AppComponent shell.
* You applied the UppercasePipe to format the name.
* You used two-way data binding with the [ngModel](https://angular.io/api/forms/NgModel) directive.
* You learned about the AppModule.
* You imported the [FormsModule](https://angular.io/api/forms/FormsModule) in the AppModule so that Angular would recognize and apply the [ngModel](https://angular.io/api/forms/NgModel) directive.
* You learned the importance of declaring components in the AppModule and appreciated that the CLI declared it for you.



Module is a collection of functionality physicly is a collection of .js file



+ go to the app.module.ts 



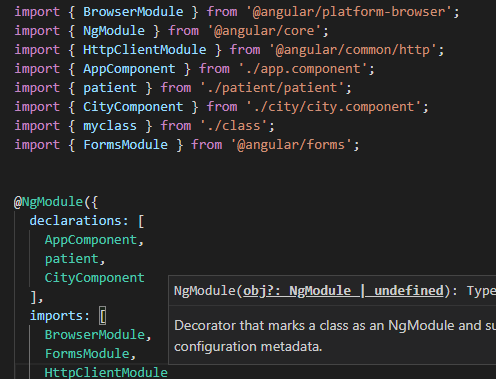
In the providers the runtime create a instance for once time and that instance was intersection with all the componant

Provide class to the servise

Imports module from the @angular

To declare the new component

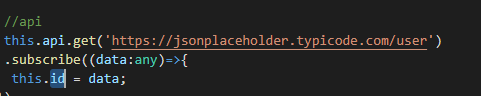
**Httpclient**



Why httpClient and not fitch?

App.component.ts 🡪 constructor:





**Step1:** Import our httpClientModule in our app module

**Step2:** import httpClient in our service

**Step3:** inject the httpClient in the constructor method of the class

**Step4:** Implement the GET method call

**Step5:** Import the service into the required calling component class

**Step6:** call the method to make the httpClient resquest

**Routing**

Routing is how you tell the angular which component you have activate by the url

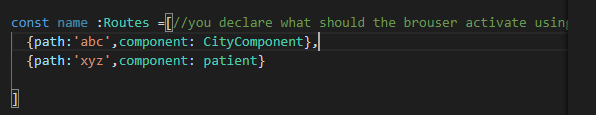
Create a const Routes (type)

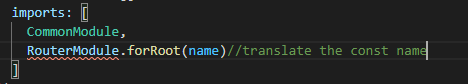
Next imports (RouterModule forRoot)

Next exports (Router module )

Next reference AppRoutingModule (class) [go to the app component 🡪 <router-outlet></router-outlet>] =>like a brize that call navigation

🡪app-routing.module.ts







There are 3 ways to navigate a component:

1. Type it by hand
2. Router link
3. service of the component

**1-type by hand**

**2-router link**

🡪app.component.html

<a routerlink='/abc'>city</a>

**3-service of the component**

Service 1: Route

* c2.component.ts

constructor( private route : Route ){}

function(){

this.route.navigate([ ‘/component’ ]); **(allowance create a property of type route after we called by this. The constructor help us to do it easily)**

}

Service 2: ActivatedRoute(to receive the information for the parent for ex the id)

We use it in the component than we wanna call it.

Example: if we wanna call c3comaponant in the maincomaponant we publish this service in c3componant.ts



Go to the OnOnit(){

const id = + this.route.snapshot.paramMap.get('id'); ( + for transformation the string to number )

**Lazy loading (on demand loading)**

It is a way to load every users to the components which he import without loading all the component

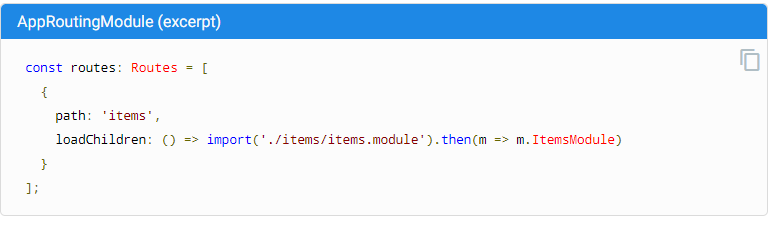
It’s a smart way to load module and related component

<https://www.digitalocean.com/community/tutorials/angular/lazy-loding>

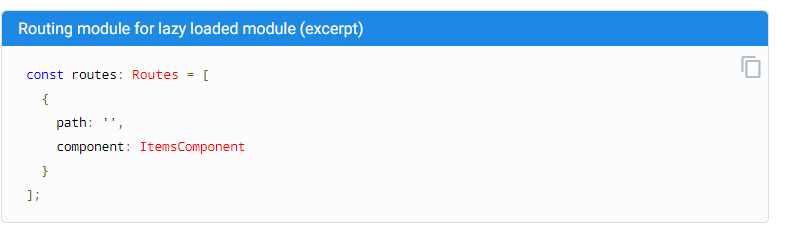
1. create a module 🡪ng g module name
2. create a component under the last module 🡪 ng g c name’/name

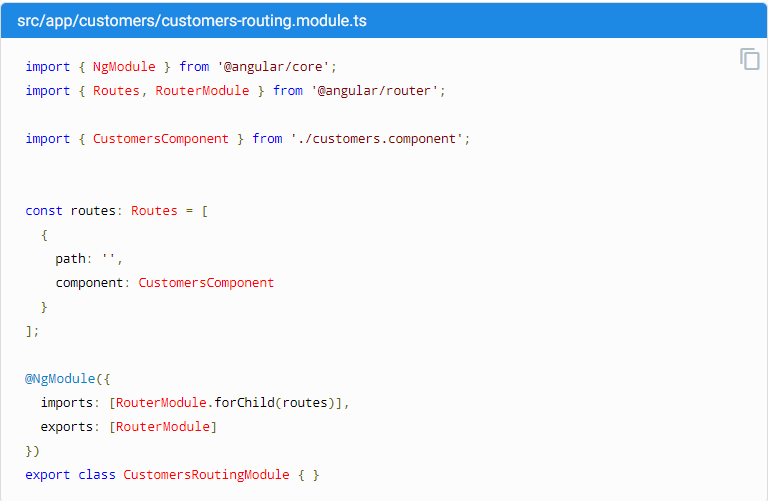
when you do that the component will declare in the name.module.ts

module is a class decorated by @ngModule



When I choosing “items” then the itemsModule activate so all component then declare inside it will activate





## forRoot() and forChild()(specific more)

You might have noticed that the CLI adds RouterModule.forRoot(routes) to the AppRoutingModule imports array. This lets Angular know that the AppRoutingModule is a routing module and forRoot() specifies that this is the root routing module. It configures all the routes you pass to it, gives you access to the router directives, and registers the [Router](https://angular.io/api/router/Router) service. Use forRoot() only once in the application, inside the AppRoutingModule.

The CLI also adds RouterModule.forChild(routes) to feature routing modules. This way, Angular knows that the route list is only responsible for providing additional routes and is intended for feature modules. You can use forChild() in multiple modules.

The forRoot() method takes care of the global injector configuration for the Router. The forChild() method has no injector configuration. It uses directives such as [RouterOutlet](https://angular.io/api/router/RouterOutlet) and [RouterLink](https://angular.io/api/router/RouterLink).

**Authentication**

Angular’s route guards are interfaces which can tell the router whether or not it should allow navigation to a requested route. They make this decision by looking for a true or false return value from a class which implements the given guard interface.

There are five different types of guards and each of them is called in a particular sequence. The router’s behavior is modified differently depending on which guard is used. The guards are:

* CanActivate
* CanActivateChild
* CanDeactivate
* CanLoad
* Resolve

1🡪create a file example.ts

// src/app/auth/auth.service.ts

import { Injectable } from '@angular/core';  
import { JwtHelperService } from '@auth0/angular-jwt';

@Injectable()  
export class AuthService {

constructor() {}

// ...  
canActivate(route: ActivatedRouteSnapshot, state: RouterStateSnapshoot): Boolean{

Return true;

}}

2🡪app-routing.module.ts

{path:’abc’,component:c1component,canActivate:[example]}

Look c1 which you activate when we write abc don’t activate else example.ts then content canActivate’s class which have canActivate function is true \

**Conditional class**

🡪test.component.ts

1-Create a property {property}:flase

2-create a function function(){

This.property =! This.property

}

🡪test.component.html

(click)=’function()’

[ngClass]="(property) ? 'my-class1':'my-class2'”

}"

If it’s false

If it’s true

Classes:

OnChanges: it is a basically a life cycle hook and it’s going to get called when something of this component directive here something inside her changes.

**Pipe**

Pipe is a filter you can create your own pipe

There are two type of pipe:

* Inbuilt pipe
* Custom pipe

**Inbuilt pipe**

Could be any one of these :

Date \ uppercase \ lowercase \ Currency \ percent \titlecase \ slice x (start with the slice nbr x)

**Custom pipe**

* Creating the custom pipe
* Transforming the value
* Returning the new value
* Accepting arguments

Example: filter.pipe.ts

**Ng g pipe pipeName**

Create a file.pipe.ts

Create a file.pipe.spec.ts

Update app.module (declare the pipe)

**\*ngFor**

1. Create the service with the array
   1. Provide the service class
   2. the array is out of the class
   3. create a property (empty array)
   4. create a function and return the property with a copy of the array

// return this.property = Array.slice( 0 );

1. Go and let this array a property in the ts file
   1. Add the private to the constructor
   2. Create a property of type empty array
   3. Define in the constructor : newproperty is equal to the implemented array

// constructor ( private newname : ClassInTheService){

This. newproperty = this. ClassInTheService.function();

}

1. Then type the code \*[ngFor](https://angular.io/api/common/NgForOf)="let name of property(in her ts file)"