Short cut to create module lazy-loading

Ng g m module\_name – route route\_name –module parent\_module\_name

Note :

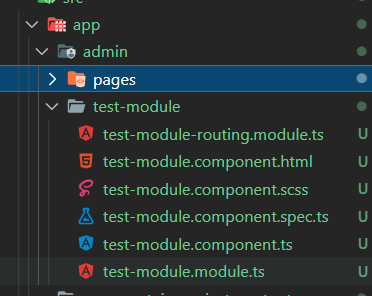
Parent\_module\_name is module that have routing so that redirect to that module\_name

Ex:

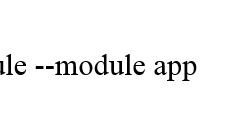
ng g m admin/test-module --route test-module --module app

Mặc định thư mục đang làm việc hiện tại ở app

Vậy nên khi ta sử dụng câu lệnh trên nó sẽ tạo ra 1 module như sau:



Sẽ có 1 điều hướng đến test-module trong app-routing.module.ts bởi vì



Tương tự ta có :

ng g m admin/pages/page1 --route page1 --module admin/test-module

trong routing của module test-module sẽ có 1 điều hướng đến module page1

**Tạo component theo module được chỉ định**

**Ng g c component\_name –module=path\_of\_modole**

For ex:

ng g c admin/pages/page1/components/sub-com –module = admin/pages/page1/page1

Ta tạo ra 1 compnent mới (sub-com) nằm trong thư mục components với module được chỉ định là page1

Convention to Angular code

<https://blogs.halodoc.io/angular-best-practices/>

Web for impove Angular:

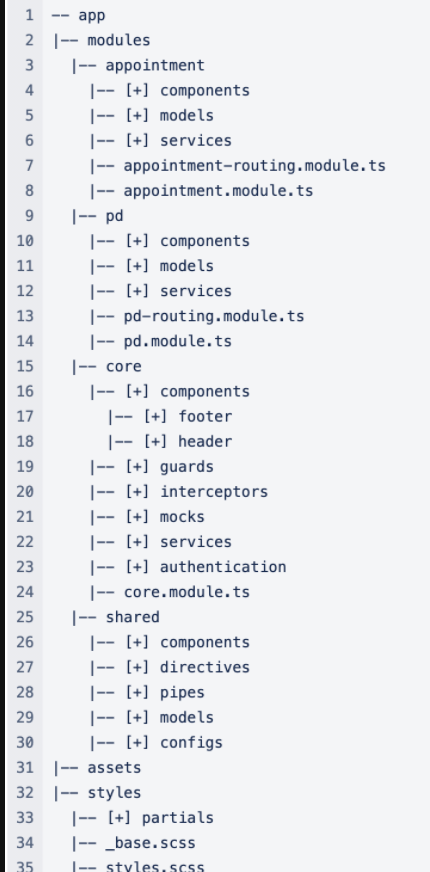
<https://blogs.halodoc.io/angular-best-practices/>

TrackBy:

<https://tharakamd-12.medium.com/avoid-unnecessary-rendering-on-angular-ngfor-with-trackby-f56ae5d79561>

Structure Project: <https://javascript.plainenglish.io/how-to-structure-angular-apps-in-2021-a0bdd481ad0d>





***Error in Angular***

Mat Form Field always attached MatInput so you must import MatInputModule with MatFormField

***FormFroup and FormArray***

FormControl is input type(<input type=”text”/>,…)

FormGroup is objects(key,value) with value is FormControl

FormArray is array contains items with item is FormControls

**Custom Validator in FormGroup**

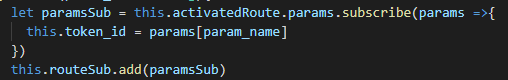


<https://www.youtube.com/watch?v=pa9S8_3Rs8A>

***Interceptor:***

Just Import HttpClientModule, HTTP\_INTERCEPTORS, APP\_INITILIZER ,… **only one** in appmodule otherwise interceptor not working

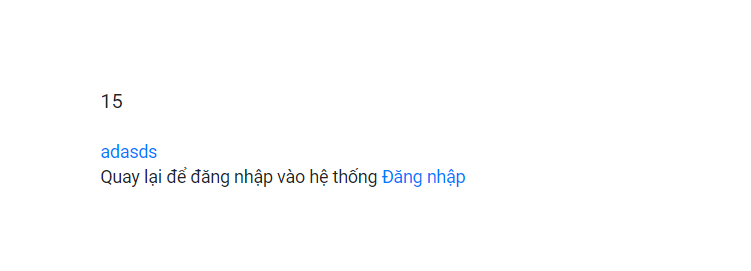
***RouteActivated***

Must use 

Otherwise router link to another will nothing happen

Just load again of that page

Like, I have a page like the following:



If click ***adasds*** will router link to the same component but different of param **id** , it will nothing happen => use **subscribe**

[**https://www.pluralsight.com/guides/accessing-route-parameters-with-activatedroute-vs.-acitivatedroutesnapshot**](https://www.pluralsight.com/guides/accessing-route-parameters-with-activatedroute-vs.-acitivatedroutesnapshot)

***Router***

Navigate used for relative path

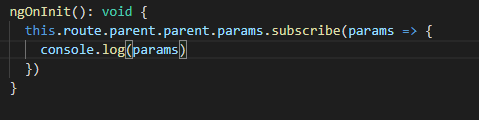
navigateByUrl used for absolute path

***this.router.navigate(['../manage'],{relativeTo:this.route})***

***Get params from parent route***

[***https://stackoverflow.com/questions/55411353/angular-get-value-of-parameter-in-parent-route***](https://stackoverflow.com/questions/55411353/angular-get-value-of-parameter-in-parent-route)

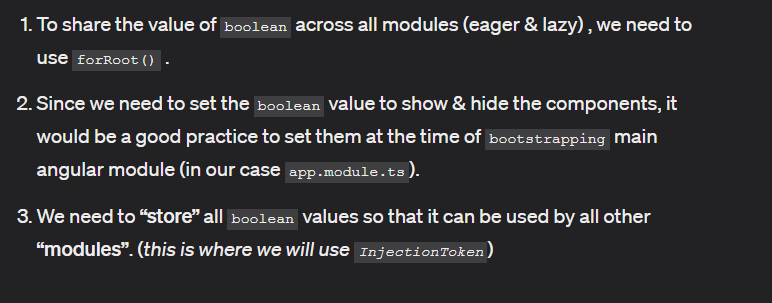
this.route.parent.(or more parent).params.subscribe((params)=>{})



***ForRoot()***

Make service through app not individual modules

***https://shashankvivek-7.medium.com/forroot-injectiontoken-in-action-angular-c2bab9959207***



***The way to create a interface with function***

social\_btns: Array<{

    icon\_path: string;

    title: string;

    click():void

  }> = [

    {

      icon\_path: 'assets/common-img/facebook-icon.jpg',

      title: 'Đăng nhập với tài khoản Facebook',

      click:()=>{

        this.FBLogin()

      }

    },

    {

      icon\_path: 'assets/common-img/google-icon.png',

      title: 'Đăng nhập với tài khoản Google',

      click:()=>{

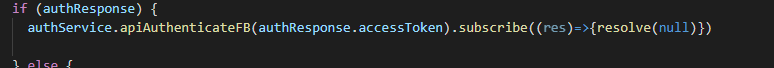
      },

    },

  ];

***Initializer file:***

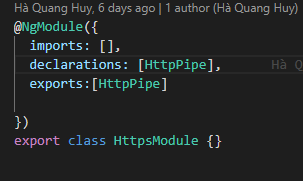
Remember add resolve at the end of function . If function is Observable resolve function is must like :



It mean wait until data back to cilent it will resolve() otherwise resolve() function will work before data back to client

***Shared componenst in single modules***

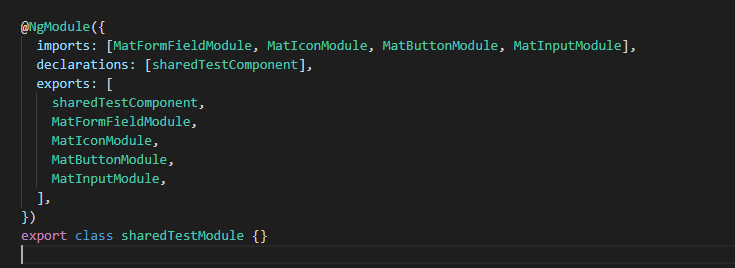
1. Chỉ có component



2.Chỉ có modules



3.Có cả component và modules



***Material***

**Add icon**

**<i class="material-icons">add\_circle\_outline</i>**

***Rich Text***

[***https://stackblitz.com/edit/angular-editor-wysiwyg-azwvpv?file=src%2Fapp%2Fapp.component.html***](https://stackblitz.com/edit/angular-editor-wysiwyg-azwvpv?file=src%2Fapp%2Fapp.component.html)

[***https://www.npmjs.com/package/@kolkov/angular-editor***](https://www.npmjs.com/package/@kolkov/angular-editor)

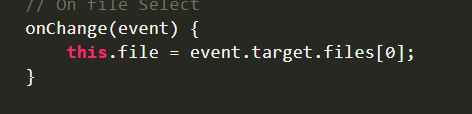
***When click into a button but reload page***

1.Look into its wrappered into a tag with href or not

2.Or button with no type or type submmit then edit type =’button’

***Upload file Angular***

******

******

only upload video

accept=".avi,.mpg,.mpeg,.flv,.mov,.m2v,.m4v,.mp4,.rm,.ram,.vob,.ogv,.webm,.wmv"

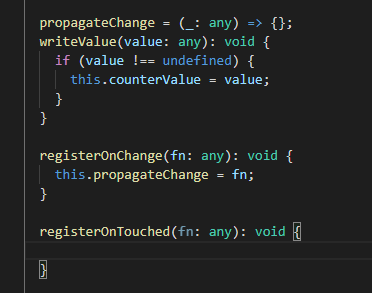
***Make ngModel on custom elements***

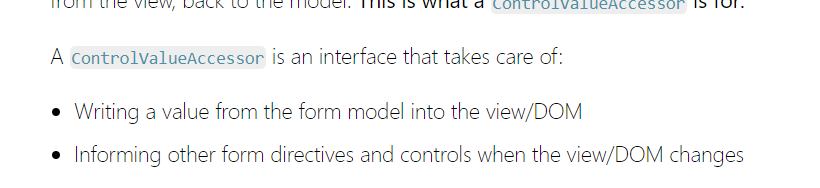
<https://blog.thoughtram.io/angular/2016/07/27/custom-form-controls-in-angular-2.html>

*Steps make custom form control:*

1. *Implement ComtrolvalueAccessor interface*

*Define all method*

**

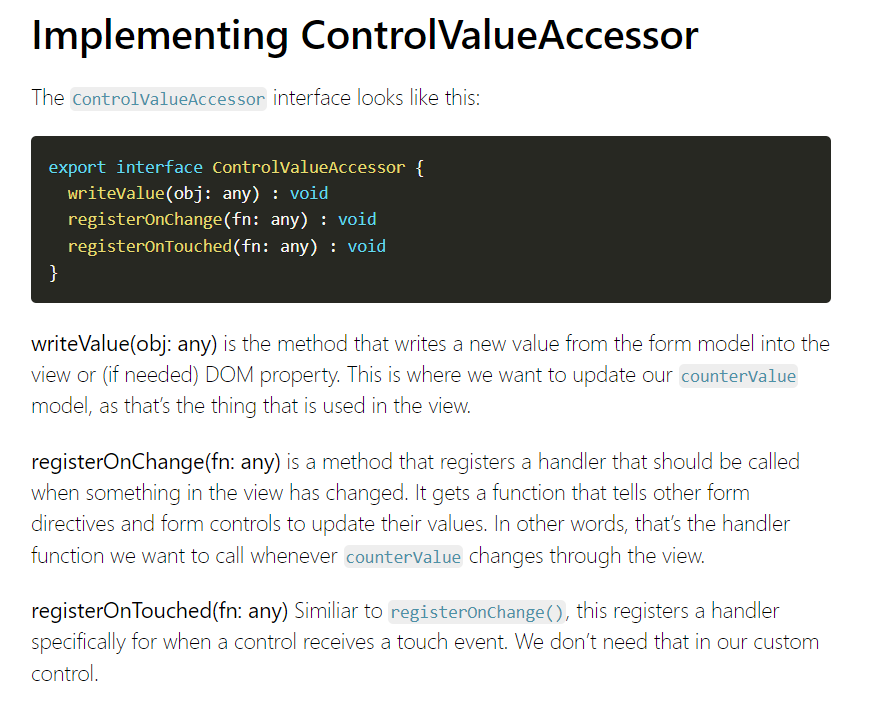
**

*Khi form được khởi động nó sẽ lần lượt vào các hàm trên*

*Chú đến hàm registerOnChange(fn) : ,fn là function được dùng để bắn ra bên ngoài các giá trị thay đổi cho thế giới biết(các directive,control khác)*

*Và ta gán hàm đó cho 1 hàm ta định nghĩa trước đó để sau này dùng hàm đó bắn ra các giá trị thay đổi(propagatedChange):*

*This.propagatedChange = fn*

**

**WriteValue is**

gets called when the form is initialized, with the form model’s initial value. This means it will override the default value 0, which is fine but if we think about the simple form setup we talked about earlier, we realise that there is no initial value in the form model:

<counter-input name="counter" ngModel></counter-input>

This will cause our component to render an empty string. As a quick fix, we only set the value when it’s not undefined:

writeValue(value: any) {

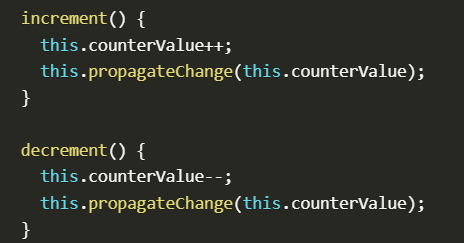
if (value !== undefined) {

this.counterValue = value;

}

}

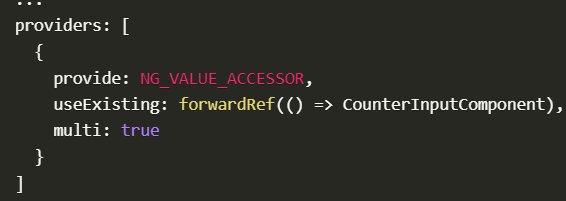
registerOnChange() has access to a function that informs the outside world about changes. Here’s where we can do special work, whenever we propagate the change, if we wanted to. registerOnTouched() registers a callback that is excuted whenever a form control is “touched”. E.g. when an input element blurs, it fire the touch event. We don’t want to do anything at this event, so we can implement the interface with an empty function.



Use propagateChange to emit the changed value to the world

1. Registering the ControlValueAccessor

 There are some DI tokens that Angular uses to inject multiple values, to do certain things with them. For example there’s the NG\_VALIDATORS token that gives Angular all registered validators on a form control, and we can add our own validators to it.



Nom na kiểu ta đăng kí đến NG\_VALUE\_ACCESSOR token để Angular hiểu CounterInputComponent được coi là 1 value accessor (1 form control)

[***Detect click outside Angular component***](https://stackoverflow.com/questions/40107008/detect-click-outside-angular-component)

[***https://stackoverflow.com/questions/40107008/detect-click-outside-angular-component***](https://stackoverflow.com/questions/40107008/detect-click-outside-angular-component)

***SEO***

[***https://www.tektutorialshub.com/angular-tutorial/#seo-angular***](https://www.tektutorialshub.com/angular-tutorial/#seo-angular)

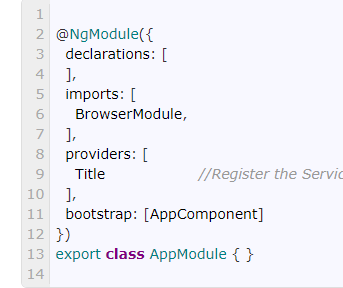
1. ***Set Page Title Using Title Service Angular Example***

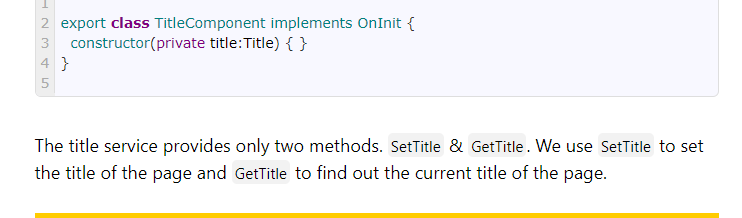
*Why change page title?*

*Changing the page title is very important as it helps the search engines to know the purpose of the page and index it properly. It also helps users to know, which page they are in.*

*Being a single page app, the angular does not reload the entire page****. The page*** *is loads only once at the startup.* ***Only part of the page*** *gets loaded when you navigate from one*[*route*](https://www.tektutorialshub.com/angular/angular-routing-navigation/)*to another.*

**

******

******

1. ***Dynamic Page Title based on Route in Angular***

***Monaco Editor***

[***https://github.com/materiahq/ngx-monaco-editor***](https://github.com/materiahq/ngx-monaco-editor)

***CRUD make observable change***

***Add:***

******

*Testcaselist is Observale that async in html*

*Add:testcaselist.push()*

*Edit:testcaselist[index]=*

*Delete:testcaselist.splice(index,numberOfDelete)*

***DI : Dependency injection***

******

DI trong Angular

Trong Angular, DI bao gồm ba thành phần sau đây:

* Injector: là một object có chứa các API để chúng ta có thể lấy về các instances đã tạo hoặc tạo các instances của các phụ thuộc.
* Provider: giống như một công thức để Injector có thể biết làm thế nào để tạo một instance của một phụ thuộc.
* Dependency: là một object (có thể là function, một value thông thường) của một kiểu dữ liệu cần phải khởi tạo.

Bạn có thể cung cấp injectors với providers ở nhiều levels khác nhau trong app, bằng một trong ba cách sau:

* Trong @Injectable() decorator cho service đó.
* Trong @NgModule() decorator (providers array) đối với NgModule.
* Trong @Component() decorator (providers array) đối với component hoặc directive (Chúng ta sẽ tìm hiểu chi tiết về Directive sau).

*Khi ta provide 1 instance of dependency như sau:*

providers: [

AuthService

  ],

*Nhưng thực ra đó là viết tắ của:*

providers: [

{

provide:AuthService,

useClass:AuthService

}

  ],

*Có nghĩa là nó sẽ tự tạo cho t 1 instance của AuthService Class*

*Note:nếu bây giờ ta muốn override AuthService class bằng 1 AuthService 2 ta làm như sau*

providers: [

{

provide:AuthService,

useClass:AuthService2

}

  ],

*Như vậy ta sẽ không cần làm như sau:*

*1.add thêm 1 provide*

providers: [

{

provide:AuthService2,

useClass:AuthService2

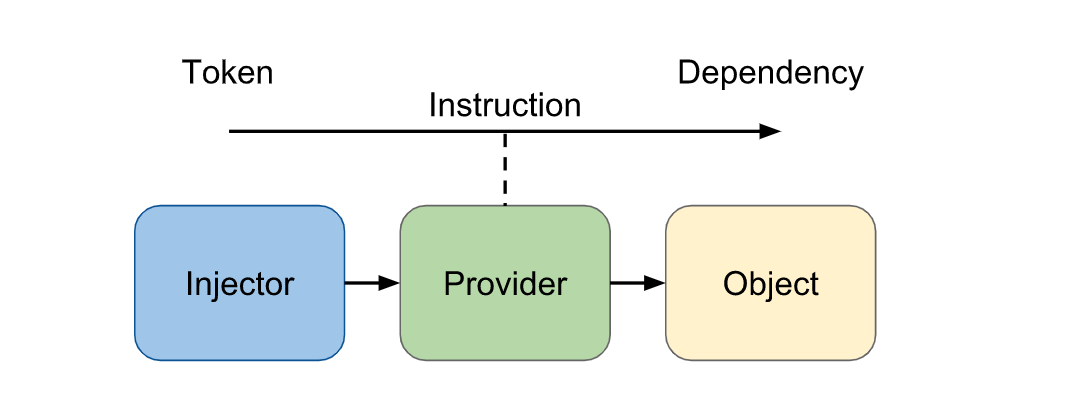
}

  ],

*Và inject AuthService2 vào contructor của component*

*Ta sẽ k cần sửa code : inject AuthService 2 vaò trong component mà vẫn sử dụng được các method của AuthService như bình thường bởi ta đã có useClass:AuthService2*

* *Những thứ ta inject vào contructor component chính là những dependency*
* *An injector creates dependencies using providers. Providers are recipes that know how to create dependencies.*

**

***Ứng dụng DI vào ActicateRoute***

[***https://github.com/angular-vietnam/100-days-of-angular/blob/master/Day048-using-dependency-injection-to-get-data-from-activated-route.md***](https://github.com/angular-vietnam/100-days-of-angular/blob/master/Day048-using-dependency-injection-to-get-data-from-activated-route.md)

***Cách tạo provide khác của service***

***Step1:***

Place a name for token

export const COURSE\_TOKEN = new InjectionToken('course-service');

***Step2:***

Procide it for angular can use it

 {

      provide: COURSE\_TOKEN,

      useClass: CourseCreationService,

    },

Note: Provide as a recipe to say to token how to create a instance of dependency(class)

***Step3:***

Use it in component

@Inject(COURSE\_TOKEN)

    private courseServices: CourseCreationService,

***Cách unsubcribe in ts file common***

destroy$: Subject<boolean> = new Subject<boolean>();

//when call api

this.courseServices

        .getCourseInfor(this.courseId, 'goals')

        .pipe(takeUntil(this.destroy$))

        .subscribe((data) => {

          this.InitArr(data);

        });

 ngOnDestroy() {

    this.destroy$.next(true);

    this.destroy$.unsubscribe();

  }