# **Application CRUD Angular 15**

1. **Creer le projet Angular :** ng new my-crud-app –routing
2. **Installer Bootstrap:** npm install bootstrap –save
3. **importer ce fichier dans « src/styles.css » :** @import "~bootstrap/dist/css/bootstrap.css";
4. **créer un module post :** ng generate module post –routing
5. **créer maintenant les composant suivantes sous le module post :**

ng generate component post/index

ng generate component post/view

ng generate component post/create

ng generate component post/edit

1. **Dans cette étape, nous allons simplement créer une route pour l'index, créer, modifier et afficher à l'aide du nouveau composant généré. nous devons donc mettre à jour notre fichier de module de post-routing comme le code ci-dessous :**

**src/app/post/post-routing.module.ts**

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

import { Routes, RouterModule } from '@angular/router';

import { IndexComponent } from './index/index.component';

import { ViewComponent } from './view/view.component';

import { CreateComponent } from './create/create.component';

import { EditComponent } from './edit/edit.component';

const routes: Routes = [

{ path: 'post', redirectTo: 'post/index', pathMatch: 'full'},

{ path: 'post/index', component: IndexComponent },

{ path: 'post/:postId/view', component: ViewComponent },

{ path: 'post/create', component: CreateComponent },

{ path: 'post/:postId/edit', component: EditComponent }

];

@NgModule({

imports: [RouterModule.forChild(routes)],

exports: [RouterModule]

})

export class PostRoutingModule { }

1. **créer une interface en utilisant la commande angular ci-dessous pour le module post. nous utiliserons l'interface post avec Observable.**

ng generate interface post/post

**src/app/post/post.ts**

export interface Post {

id: number;

title: string;

body: string;

}

1. **Génerer un service de publication et mettons tout le code pour la méthode de service Web :** ng generate service post/post

**src/app/post/post.service.ts**

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

import { HttpClient, HttpHeaders } from '@angular/common/http';

import { Observable, throwError } from 'rxjs';

import { catchError } from 'rxjs/operators';

import { Post } from './post';

@Injectable({

providedIn: 'root'

})

export class PostService {

private apiURL = "https://jsonplaceholder.typicode.com";

httpOptions = {

headers: new HttpHeaders({

'Content-Type': 'application/json'

})

}

constructor(private httpClient: HttpClient) { }

getAll(): Observable<any> {

return this.httpClient.get(this.apiURL + '/posts/')

.pipe(

catchError(this.errorHandler)

)

}

create(post:Post): Observable<any> {

return this.httpClient.post(this.apiURL + '/posts/', JSON.stringify(post), this.httpOptions)

.pipe(

catchError(this.errorHandler)

)

}

find(id:number): Observable<any> {

return this.httpClient.get(this.apiURL + '/posts/' + id)

.pipe(

catchError(this.errorHandler)

)

}

update(id:number, post:Post): Observable<any> {

return this.httpClient.put(this.apiURL + '/posts/' + id, JSON.stringify(post), this.httpOptions)

.pipe(

catchError(this.errorHandler)

)

}

delete(id:number){

return this.httpClient.delete(this.apiURL + '/posts/' + id, this.httpOptions)

.pipe(

catchError(this.errorHandler)

)

}

errorHandler(error:any) {

let errorMessage = '';

if(error.error instanceof ErrorEvent) {

errorMessage = error.error.message;

} else {

errorMessage = `Error Code: ${error.status}\nMessage: ${error.message}`;

}

return throwError(errorMessage);

}

}

1. **Mettre à jour la logique et le modèle de composant**

**src/app/post/index/index.component.ts**

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

import { PostService } from '../post.service';

import { Post } from '../post';

@Component({

selector: 'app-index',

templateUrl: './index.component.html',

styleUrls: ['./index.component.css']

})

export class IndexComponent implements OnInit {

posts: Post[] = [];

constructor(public postService: PostService) { }

ngOnInit(): void {

this.postService.getAll().subscribe((data: Post[])=>{

this.posts = data;

console.log(this.posts);

})

}

deletePost(id:number){

this.postService.delete(id).subscribe(res => {

this.posts = this.posts.filter(item => item.id !== id);

console.log('Post deleted successfully!');

})

}

}

1. **Mettre à jour la logique et le modèle de composant**

**src/app/post/index/index.component.html**

<div class="container">

<h1>Angular 15 CRUD Example - ItSolutionStuff.com</h1>

<a href="#" routerLink="/post/create/" class="btn btn-success">Create New Post</a>

<table class="table table-striped">

<thead>

<tr>

<th>ID</th>

<th>Title</th>

<th>Body</th>

<th width="250px">Action</th>

</tr>

</thead>

<tbody>

<tr \*ngFor="let post of posts">

<td>{{ post.id }}</td>

<td>{{ post.title }}</td>

<td>{{ post.body }}</td><td>

<a href="#" [routerLink]="['/post/', post.id, 'view']" class="btn btn-info">View</a>

<a href="#" [routerLink]="['/post/', post.id, 'edit']" class="btn btn-primary">Edit</a>

<button type="button" (click)="deletePost(post.id)" class="btn btn-danger">Delete</button>

</td></tr>

</tbody>

</table>

</div>

**src/app/post/create/create.component.ts**

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

import { PostService } from '../post.service';

import { Router } from '@angular/router';

import { FormGroup, FormControl, Validators} from '@angular/forms';

@Component({

selector: 'app-create',

templateUrl: './create.component.html',

styleUrls: ['./create.component.css']

})

export class CreateComponent implements OnInit {

form!: FormGroup;

constructor(

public postService: PostService,

private router: Router

) { }

ngOnInit(): void {

this.form = new FormGroup({

title: new FormControl('', [Validators.required]),

body: new FormControl('', Validators.required)

});

}

get f(){

return this.form.controls;

}

submit(){

console.log(this.form.value);

this.postService.create(this.form.value).subscribe((res:any) => {

console.log('Post created successfully!');

this.router.navigateByUrl('post/index');

})

}

}

**src/app/post/create/create.component.html**

<div class="container">

<h1>Create New Post</h1>

<a href="#" routerLink="/post/index" class="btn btn-primary">Back</a>

<form [formGroup]="form" (ngSubmit)="submit()">

<div class="form-group">

<label for="title">Title:</label>

<input

formControlName="title"

id="title"

type="text"

class="form-control">

<div \*ngIf="f['title'].touched && f['title'].invalid" class="alert alert-danger">

<div \*ngIf="f['title'].errors && f['title'].errors['required']">Title is required.</div>

</div>

</div>

<div class="form-group">

<label for="body">Body</label>

<textarea

formControlName="body"

id="body"

type="text"

class="form-control">

</textarea>

<div \*ngIf="f['body'].touched && f['body'].invalid" class="alert alert-danger">

<div \*ngIf="f['body'].errors && f['body'].errors['required']">Body is required.</div>

</div>

</div>

<button class="btn btn-primary" type="submit" [disabled]="!form.valid">Submit</button>

</form>

</div>

**src/app/post/edit/edit.component.ts**

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

import { PostService } from '../post.service';

import { ActivatedRoute, Router } from '@angular/router';

import { Post } from '../post';

import { FormGroup, FormControl, Validators} from '@angular/forms';

@Component({

selector: 'app-edit',

templateUrl: './edit.component.html',

styleUrls: ['./edit.component.css']

})

export class EditComponent implements OnInit {

id!: number;

post!: Post;

form!: FormGroup;

constructor(

public postService: PostService,

private route: ActivatedRoute,

private router: Router

) { }

ngOnInit(): void {

this.id = this.route.snapshot.params['postId'];

this.postService.find(this.id).subscribe((data: Post)=>{

this.post = data;

});

this.form = new FormGroup({

title: new FormControl('', [Validators.required]),

body: new FormControl('', Validators.required)

});

}

get f(){

return this.form.controls;

}

submit(){

console.log(this.form.value);

this.postService.update(this.id, this.form.value).subscribe((res:any) => {

console.log('Post updated successfully!');

this.router.navigateByUrl('post/index');

})

}}

**src/app/post/edit/edit.component.html**

<div class="container">

<h1>Update Post</h1>

<a href="#" routerLink="/post/index" class="btn btn-primary">Back</a>

<form [formGroup]="form" (ngSubmit)="submit()">

<div class="form-group">

<label for="title">Title:</label>

<input

formControlName="title"

id="title"

type="text"

[(ngModel)]="post.title"

class="form-control">

<div \*ngIf="f['title'].touched && f['title'].invalid" class="alert alert-danger">

<div \*ngIf="f['title'].errors && f['title'].errors['required']">Title is required.</div>

</div>

</div>

<div class="form-group">

<label for="body">Body</label>

<textarea

formControlName="body"

id="body"

type="text"

[(ngModel)]="post.body"

class="form-control">

</textarea>

<div \*ngIf="f['body'].touched && f['body'].invalid" class="alert alert-danger">

<div \*ngIf="f['body'].errors && f['body'].errors['required']">Body is required.</div>

</div>

</div>

<button class="btn btn-primary" type="submit" [disabled]="!form.valid">Update</button>

</form>

</div>

**src/app/post/view/view.component.ts**

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

import { PostService } from '../post.service';

import { ActivatedRoute, Router } from '@angular/router';

import { Post } from '../post';

@Component({

selector: 'app-view',

templateUrl: './view.component.html',

styleUrls: ['./view.component.css']

})

export class ViewComponent implements OnInit {

id!: number;

post!: Post;

constructor(

public postService: PostService,

private route: ActivatedRoute,

private router: Router

) { }

ngOnInit(): void {

this.id = this.route.snapshot.params['postId'];

this.postService.find(this.id).subscribe((data: Post)=>{

this.post = data;

});

}}

**src/app/post/view/view.component.html**

<div class="container">

<h1>View Post</h1>

<a href="#" routerLink="/post/index" class="btn btn-primary">Back</a>

<div>

<strong>ID:</strong>

<p>{{ post.id }}</p>

</div>

<div>

<strong>Title:</strong>

<p>{{ post.title }}</p>

</div>

<div>

<strong>Body:</strong>

<p>{{ post.body }}</p>

</div>

</div>

**src/app/app.component.html**

<router-outlet></router-outlet>

**src/app/app.module.ts**

import { BrowserModule } from '@angular/platform-browser';

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

import { HttpClientModule } from '@angular/common/http';

import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';

import { PostModule } from './post/post.module';

@NgModule({

declarations: [

AppComponent

],

imports: [

BrowserModule,

AppRoutingModule,

PostModule,

HttpClientModule

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

**src/app/post/post.module.ts**

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

import { CommonModule } from '@angular/common';

import { PostRoutingModule } from './post-routing.module';

import { IndexComponent } from './index/index.component';

import { ViewComponent } from './view/view.component';

import { CreateComponent } from './create/create.component';

import { EditComponent } from './edit/edit.component';

import { FormsModule, ReactiveFormsModule } from '@angular/forms';

@NgModule({

declarations: [IndexComponent, ViewComponent, CreateComponent, EditComponent],

imports: [

CommonModule,

PostRoutingModule,

FormsModule,

ReactiveFormsModule

]

})

export class PostModule { }

1. **Executer l’application :** ng serve

http://localhost:4200/post