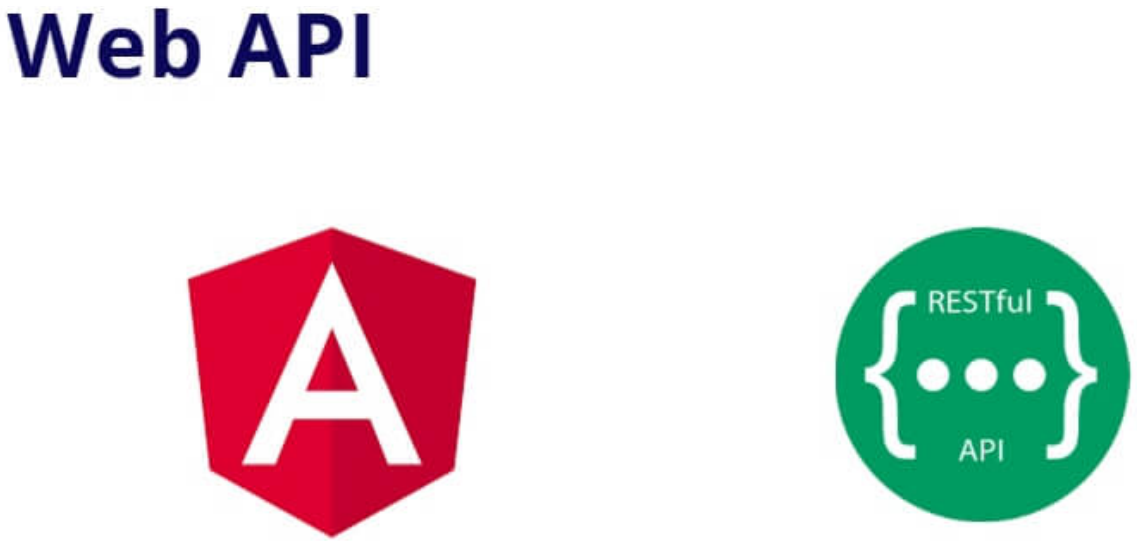

Angular 13 CRUD Example with Web API

June 12, 2022 | By Admin | 2 Comments

Use the following steps and create crud (create, read, update, delete) app in angular 13 apps; as follows:

- Step 1 – Create New Angular App
- Step 2 – Install Bootstrap
- Step 3 – Create Module & Routing
- Step 4 – Create CRUD Component For Module
- Step 5 – Adding Routes
- Step 6 – Create Interface
- Step 7 – Create Services
- Step 8 – Add Code in Component and Template
- Step 9 – Import to Modules
- Step 10 – Start Angular App

Step 1 – Create New Angular App

First of all, open your terminal and execute the following command on it to install angular app:

```
ng new my-crud-app --routing
```

```
npm install bootstrap --save
```

After that, import this css file into angular crud app:

src/styles.css

Step 3 – Create Module & Routing

In this step, execute the following command on terminal to create module in angular app:

```
ng generate module post --routing
```

The above given command will create files in the following locations:

> Angular 13 Material Datepicker Example

Step 4 – Create CRUD Component

In this step, execute the following commands on the terminal to create CRUD components in angular app:

```
ng generate component post/index
ng generate component post/view
ng generate component post/create
ng generate component post/edit
```

Now, you can see that, the above commands will create some component files into the following locations:

```
src/app/post/index/*
src/app/post/view/*
```

src/app/post/create/*

Step 5 – Adding Routes

In this step, visit the **src/app/post directory and open post-routing.module.ts** Then add the routes into **post-routing.module.ts** file:

```
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({
```

```
17    imports: [RouterModule.forChild(routes)],
18    exports: [RouterModule]
```

> Angular 13 Image Upload with Preview Example

Step 6 – Create Interface

Now, execute the following command on terminal to create interface:

```
ng generate interface post/post
```

Then visit **src/app/post directory and open post.ts**. And add the following code into it:

```
export interface Post {
id: number;
title: string;
body: string;
}
```

Step 7 – Create Services

```
ng generate service post/post
```

After that, visit **src/app/post directory and open post.service.ts**. Then add the following code into it:

```
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<Post[]> {
```

```
25      return this.httpClient.get<Post[]>(this.apiURL + '/posts/')
26      .pipe(
```

```
35      )
36    }
37
38    find(id): Observable<Post> {
39      return this.httpClient.get<Post>(this.apiURL + '/posts/' + id)
40      .pipe(
41        catchError(this.errorHandler)
42      )
43    }
44
45    update(id, post): Observable<Post> {
46      return this.httpClient.put<Post>(this.apiURL + '/posts/' + id, JSON.stringify(post), this.httpOptions)
47      .pipe(
48        catchError(this.errorHandler)
49      )
50    }
51
52    delete(id){
53      return this.httpClient.delete<Post>(this.apiURL + '/posts/' + id, this.httpOptions)
54      .pipe(
55        catchError(this.errorHandler)
56      )
57    }
58
59
60    errorHandler(error) {
61      let errorMessage = '';
62      if(error.error instanceof ErrorEvent) {
63        errorMessage = error.error.message;
64      } else {
65        errorMessage = `Error Code: ${error.status}\nMessage: ${error.message}`;
66      }
67      return throwError(errorMessage);
68    }
69  }
```

Note that, In this example, we will use https://jsonplaceholder.typicode.com web site api. If you want to create Apis or web services in PHP, Laravel and codeigniter, so you can click on the following urls:

- Create Web Service using PHP Laravel
- Create Web Service using PHP Codeigniter

> Angular 13 File Upload Example

Step 8 – Add Code in Component and Template

In this step, Create components and template like list page, create page, edit page and view page in angular crud app.

1) List Page Template and Component

```
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[] = [];

  /*----------------------------------------
```

```
    ------------------------------------------
    Created constructor
```

```
     */
    ngOnInit(): void {
      this.postService.getAll().subscribe((data: Post[])=>{
        this.posts = data;
        console.log(this.posts);
      })
    }

    /**
     * Write code on Method
     *
     * @return response()
     */
    deletePost(id:number){
      this.postService.delete(id).subscribe(res => {
           this.posts = this.posts.filter(item => item.id !== id);
           console.log('Post deleted successfully!');
      })
    }

}
```

Now, visit **src/app/post/index** directory and open **index.component.html** or create index.component.html, if not exist. Then add the following code into it:

```
<div class="container">
```

```
    <h1>Angular 13 CRUD Example - Tutsmake.com</h1>
```

```
      </tr>
      <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>
    </table>
  </div>
```

2) Add Post Page Template and Component

First of all, visit **src/app/post/create** directory and open **create.component.ts** or create create.components.ts, if not exist. Then add the following code into it:

```
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',
```

```
  ------------------------------------------
  Created constructor
  ------------------------------------------
  ------------------------------------------*/
  constructor(
    public postService: PostService,
    private router: Router
  ) { }

  /**
   * Write code on Method
   *
   * @return response()
   */
  ngOnInit(): void {
    this.form = new FormGroup({
      title: new FormControl('', [Validators.required]),
      body: new FormControl('', Validators.required)
    });
  }

  /**
   * Write code on Method
   *
   * @return response()
   */
  get f(){
    return this.form.controls;
  }

  /**
   * Write code on Method
   *
   * @return response()
   */
  submit(){
```

```
52        console.log(this.form.value);
53        this.postService.create(this.form.value).subscribe((res:any) => {
```

Now, visit **src/app/post/create a** directory and open **create.component.html** or create create.component.html, if not exist. Then add the following code into it:

```
1  <div class="container">
2      <h1>Create New Post</h1>
3
4      <a href="#" routerLink="/post/index" class="btn btn-primary">Back</a>
```

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

3) Edit Page Template and Component

Next, visit **src/app/post/edit** directory **and open edit.component.ts** or create edit.components.ts, if not exist. Then add the following code into it:

```
  selector: 'app-edit',
  templateUrl: './edit.component.html',
  styleUrls: ['./edit.component.css']
})
export class EditComponent implements OnInit {

  id!: number;
  post!: Post;
  form!: FormGroup;

  /*-----------------------------------------
  -----------------------------------------
  Created constructor
  -----------------------------------------
  -----------------------------------------*/
  constructor(
    public postService: PostService,
    private route: ActivatedRoute,
    private router: Router
  ) { }

  /**
   * Write code on Method
   *
   * @return response()
   */
  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)
    });
```

```
    }

  /**
   * Write code on Method
   *
   * @return response()
   */
  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');
    })
  }

}
```

Now, visit **src/app/post/edit a** directory and open **edit.component.html** or create edit.component.html, if not exist. Then add the following code into it:

```
<div class="container">
    <h1>Update Post</h1>
```

```
                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>
```

4) Detail Page Template and Component

Visit **src/app/post/view** directory and open **view.component.ts** or view.components.ts, if not exist. Then add the following code into it:

```
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;

  /*-----------------------------------------
  -----------------------------------------
  Created constructor
  -----------------------------------------
  -----------------------------------------*/
  constructor(
    public postService: PostService,
    private route: ActivatedRoute,
    private router: Router
   ) { }

  /**
   * Write code on Method
   *
   * @return response()
   */
  ngOnInit(): void {
    this.id = this.route.snapshot.params['postId'];

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

```

```
}
```

```
<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>
```

Now, visit **src/app directory and open app.component.html**. Then add the following line into it:

In this step, Now, visit **src/app directory and open app.module.ts**. Then add the following line into it:

```
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 { }
```

And visit **src/app/post directory and open post.module.ts**. Then add the following line into it:

```
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 { }
```

Step 10 – Start Angular App

In this step, execute the following command on terminal to start angular app:

```
ng serve
```

Then open your browser and hit the following url on it:

```
localhost:4200/post
```

Filed Under: Angular

Admin

My name is Devendra Dode. I am a full-stack developer, entrepreneur, and owner of Tutsmake.com. I like writing tutorials and tips that can help other developers. I share tutorials of PHP, Python, Javascript, JQuery, Laravel, Livewire, Codeigniter, Node JS, Express JS, Vue JS, Angular JS, React Js, MySQL, MongoDB, REST APIs, Windows, Xampp, Linux, Ubuntu, Amazon AWS, Composer, SEO, WordPress, SSL and Bootstrap from a starting stage. As well as demo example.

View all posts by Admin

PREVIOUS

Angular 13 Routing Example

2 replies to Angular 13 CRUD Example with Web API

sarita

2 months ago

very good example

Reply

Prabhakaran

2 months ago

Excellent example angular

Curd operation

Reply

Name *

Email *

Website

Save my name, email, and website in this browser for the next time I comment.

POST COMMENT

X

Angular 14 HttpClient & Http Services Example Tutori

Angular 14 Server Side Pagination Tutorial

Angular 14 Routing Example Tutorial

How to Add and Remove Users on Ubuntu 22.04 Command Line

Categories

Laravel

MySQL

NGINX

Node js

PHP

Python

React

REST API

SEO

Snowflake

SQLite

SSL Certificate

Uncategorized

Vue

Windows

Wordpress

Xampp

Recent Comments

Anshuman on C Program to Check Neon Number

Contact Form on PHP Code: Insert Data Into MySQL Database From Form

Ravi on How to Set Up File Permissions in Laravel 8

Best Free Seo Analysis Tools Best Free Seo Tools Calculate Days,Hour Betw een Tw o Dates in MySQL Query Closest() Method By jQuery codeigniter 4 example Codeigniter Ajax Image Store Into Database Codeigniter Ajax Load More Page Scroll Live Demo Codeigniter Crop Image Before Upload using jQuery Ajax Codeigniter Crud Tutorial With Source Code Codeigniter First Crud With Live Demo Codeigniter Image Crop Resize and Save Codeigniter Send Email From Localhost Xampp How-to-Install Laravel on Windows with Composer How to Import Export Excel in Laravel How to Install Apache in Ubuntu 18.04 How to Laravel Image Upload in Database How to Make User Login and Registration Laravel How to Send Mail in Laravel Install PHP in Ubuntu from Scratch Laravel 5 Send Email Example Laravel Create Update Delete Example Laravel Import Excel to Database Laravel Import Export Excel to Database Example Laravel Login Authentication Using Email Tutorial Laravel Mail Send with Example Laravel Send Email with Example Laravel Upload Image Using Ajax Tutorial Mail Send in Laravel with Example Sending Email Via Gmail SMTP Server In Laravel Send Mail PHP Laravel Example Step by Step Guide to Building Your First Laravel Application Stripe Payement Gateway Integration in Laravel

Word & Character Counter

Compare Code & Text Files

Youtube Video Downloader