CRUD PRACTICE USING HTTP REQUEST

How to make TO DO LIST

A: First, create a “todos.json” inside asset folder, this will serve to be your database.

```
{
  "todos": [
    {
      "id": 1,
      "title": "Take a Bath"
    },
    {
      "id": 2,
      "title": "HALA"
    },
    {
      "id": 3,
      "title": "KUmain"
    }
  ]
}
```

After putting the code, go to that directory and run this on terminal using the code => “json-server –p 3000 todos.json”.

B: Your database is now running on Localhost:3000/todos. Next is to create a model and service of the database.

MODEL : todo.model.ts

```
export class Todo {
    id:number;
    title:string;
}
```

SERVICE: todo.service.ts

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Todo } from './todo.model';
import { Observable } from 'rxjs/Observable';
import { Response } from '@angular/http';
import { FormGroup, FormControl } from '@angular/forms';

@Injectable({
  providedIn: 'root'
})
export class TodoService {

  public dbUrl:string = "http://localhost:3000/todos"

  id:number;
  todoObj = {};
```

```
  toDo:FormGroup;

  constructor(private http:HttpClient) {
    this.toDo = new FormGroup({
      'id': new FormControl,
      'title': new FormControl
    })
  }

  //dapat nasa format siya ng model
  getToDoList():Observable<Todo[]>{
    return this.http.get<Todo[]>(this.dbUrl);
  }

  //add
  addToDo(todo){
    this.todoObj = {
      'id':todo.id,
      'title':todo.title
    }
    return this.http.post(this.dbUrl,this.todoObj).subscribe(
      (res:Response)=>{
        console.log(res);
      }
    );
  }

  //edit
  populateForm(todo){
    this.toDo.setValue(todo);
  }
  editToDo(todo){
    this.todoObj = {
      'id':todo.id,
      'title':todo.title
    }
    return this.http.put(`${this.dbUrl}/${todo.id}`,this.todoObj).subscribe(
      (res:Response)=>{
        console.log(res)
      }
    )
  }

  //delete
  deleteToDo(id:number): Observable<void>{
    return this.http.delete<void>(`${this.dbUrl}/${id}`)
  }
}
```

C: Make the Components

Adding Component : add-todo.component.css|.html|.ts

```
<form [formGroup]="toDoForm" (ngSubmit)="onSubmit()">
  <input type="hidden" formControlName="id">
  <input class="form-control" placeholder="What to Do?" type="text" formControlName="title">
  <button class="btn btn-success" type="submit">Submit</button>
</form>
```

```
import { Component, OnInit } from '@angular/core';
import { TodoService } from 'src/app/shared/todo.service';
import { Response } from '@angular/http';

@Component({
  selector: 'app-add-todo',
  templateUrl: './add-todo.component.html',
  styleUrls: ['./add-todo.component.css']
})
export class AddTodoComponent implements OnInit {

  constructor(private todoService:TodoService) { }

  toDoForm = this.todoService.toDo

  ngOnInit() {
  }

  onSubmit(){
    if(this.toDoForm.get('id').value === null){
      this.todoService.addToDo(this.toDoForm.value)
    }else{
      this.todoService.editToDo(this.toDoForm.value)
    }

    this.toDoForm.reset()
  }

}
```

To Do List Component : todo-list.component.css|.html|.ts

```
<table class="table table-hover table-striped">
  <thead>
    <tr>
      <td>ID</td>
      <td>Title</td>
      <td>Actions</td>
    </tr>
  </thead>
  <tbody>
    <tr *ngFor="let todo of todos">
      <td>{{todo.id}}</td>
      <td>{{todo.title}}</td>
      <td>
        <button class="btn btn-primary" (click)="onEdit(todo)">Edit</button>
        <button class="btn btn-danger"(click)="onDelete(todo.id)">Delete</button>
      </td>
    </tr>
  </tbody>
</table>
```

```
import { Component, OnInit } from '@angular/core';
import { TodoService } from 'src/app/shared/todo.service';

@Component({
  selector: 'app-todo-list',
  templateUrl: './todo-list.component.html',
  styleUrls: ['./todo-list.component.css']
})
export class TodoListComponent implements OnInit {

  public todos = [];

  constructor(private todoService:TodoService) { }

  ngOnInit() {
    this.todoService.getToDoList().subscribe(
      data => this.todos = data
    )
  }

  onEdit(todo){
    this.todoService.populateForm(todo);
  }

  onDelete(id:number){
    this.todoService.deleteToDo(id).subscribe(
```

```
      ()=>console.log(`Todo with id = ${id} deleted`),
      (err)=> console.log(err)
    )
  }
}
```

D: AppModule: don't forget to import HttpClient, Service, etc

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
//import { FormsModule } from '@angular/forms';
import { ReactiveFormsModule, FormsModule } from '@angular/forms';
import { HttpModule } from '@angular/http';
//for http request
import { HttpClientModule } from '@angular/common/http';
//for routing
import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';
import { ToDoListComponent } from './to-do-list/to-do-list.component';
import { AddTodoComponent } from './to-do-list/add-todo/add-todo.component';
import { TodoListComponent } from './to-do-list/todo-list/todo-list.component';



@NgModule({
  declarations: [
    AppComponent,
    ToDoListComponent,
    AddTodoComponent,
    TodoListComponent,
  ],
  imports: [
    BrowserModule,
    FormsModule,
    ReactiveFormsModule,
    HttpModule,
    AppRoutingModule,
    HttpClientModule,
  ],
  providers: [TodoService],
  bootstrap: [AppComponent]
})
export class AppModule { }
```