Generalitat de Catalunya
Consorci per a la Formació Contínua de Catalunya

pime Coneixement

MINISTERIO DE TRABAJO, MIGRACIONES Y SEGURIDAD SOCIAL
SERVICIO PÚBLICO DE EMPLEO ESTATAL

Angular
IFCD65

Formador: Borja Mulleras Vinzia

CRUD - ANGULAR

En base al ejemplo anterior y con la API rest que hemos utilizado, vamos a preparar un CRUD de angular

Consultar detalle del TODO

Modificamos el proyecto Serveis

PASO 1

- Creamos una propiedad del proyecto con la URL de la API Rest que consultamos. Modificamos todo.service.ts

CRUD - ANGULAR

```
import { Injectable } from '@angular/core';
import { Todo } from 'src/app/modelos/todo';
import { Observable } from 'rxjs';
import { HttpClient } from '@angular/common/http';
@Injectable({
  providedIn: 'root'
})
export class TodoService {
  public api:string;
  constructor(private http:HttpClient) {
    //Recurso a utilizar
    this.api= 'https://jsonplaceholder.typicode.com/todos/';
  }
  //método para consultar todas las tareas
  public listaTareas():Observable<Todo[]> {
      //peticion asincrona al servicio
      return this.http.get<Todo[]>(this.api);
    }
}
```

CRUD - ANGULAR

Y añadimos el método de detalleTarea en el mismo todo.service.ts

```
//método para consultar la tarea con el id correspondiente
 public detalleTarea(id?: number):Observable<Todo> {
    //peticion asincrona al servicio ---> todos/id
    return this.http.get<Todo>(`${this.api}/${id}`);
 }
```

CRUD - ANGULAR

PASO 2

Añadimos a todo.component.ts la función de consultaTarea

```
public consultaTarea() {
      let tarea = this.servicio.detalleTarea(this.id).subscribe({
      //función a ejecutar si la petición ha ido bien
      next: (todos) => {
                        console.log(todos);
                        this.tarea=todos;
                      },
      //función a ejecutar si hay un error
      error: (error) => console.log('Algo fue mal',error),
      //funcion a ejecutar si la tarea se completa
      complete: () => console.log('Tarea completada')
      })
  }
```

CRUD - ANGULAR

PASO 2 Bis
También cambiamos el constructor y la definición de propiedades

```
public listaTareas?:Todo[];
   public tarea?:Todo;
   public id?:number;
   constructor (private servicio:TodoService,private rutaActiva:ActivatedRoute) {
       this.listaTareas=undefined;
       this.tarea=undefined;
       this.id = this.rutaActiva.snapshot.params['id'];
       this.consultaTareas();
       this.consultaTarea();
   }
```

CRUD - ANGULAR

PASO 3

En todo.component.html crearemos 2 plantillas, una para mostrar el listado de tareas y otra para mostrar la info especifica de la tarea

```
<span *ngIf="tarea==undefined; then listTareas else tareaUnica"></span>
<ng-template #listTareas>
<h1>Lista de tareas</h1>
<div>
        <div *ngFor="let t of listaTareas">
                <span class="id" >{{t.id}} </span>
                <span class="tarea">
                        <a [routerLink]="['/listaTareas',t.id]">
                                {{t.title}}
                        </a>
                </span>
        </div>
</div>
</ng-template>
```

CRUD - ANGULAR

PASO 3

En todo.component.html crearemos 2 plantillas, una para mostrar el listado de tareas y otra para mostrar la info especifica de la tarea

```
<ng-template #tareaUnica>
<h1>Tarea</h1>
<div *ngIf="tarea">
            <span class="id" >{{tarea.id}} </span>
            <span class="tarea">{{tarea.title}}</span>
</div>
</ng-template>
```

CRUD - ANGULAR

PASO 4 Alta Tarea
Modificamos todo.service.ts añadiendo el método de alta

```
//método para añadir la tarea pasada por parametro
 public postTarea(tarea:Todo):Observable<Todo>{
    return this.http.post<Todo>(this.api,tarea);
 }
```

CRUD - ANGULAR

PASO 5 Alta Tarea

Modificamos todo.component.ts añadiendo el método de alta

```
public altaTarea() {
      let tarea:Todo= {
          userId: 2,
          title: "nueva tarea añadida por el programa de Borja",
          completed: false
      }
      this.servicio.postTarea(tarea).subscribe({
      //función a ejecutar si la petición ha ido bien
      next: (todos) => {
                        console.log(todos);
                        this.tarea=todos;
                      },
      //función a ejecutar si hay un error
      error: (error) => console.log('Algo fue mal',error),
      //funcion a ejecutar si la tarea se completa
      complete: () => console.log('Tarea completada')
      })
  }
```

CRUD - ANGULAR

PASO 6 Alta Tarea
Modificamos todo.component.html añadiendo al final un botón para ejecutar el alta y comprobar que funciona

```
…
<button (click)="altaTarea();">Alta Tarea</button>
<pre>
        {{tarea | json}}
</pre>
```

CRUD - ANGULAR

PASO 7 Modificar Tarea
Modificamos todo.service.ts añadiendo el método de modificar

```
//método para modificar una tarea
 public putTarea(tarea:Todo):Observable<Todo>{
     //El primer parametro es la URL del todo a modificar y
     //el segundo es la tarea
     return this.http.put<Todo>(`${this.api}/${tarea.id}`,tarea);
 }
```

CRUD - ANGULAR

PASO 8 Modificar Tarea

Modificamos todo.component.ts añadiendo el método de modificacion

```
public modificarTarea() {
      let tarea:Todo= {
          userId: 3,
          id: 1,
          title: "Tarea modficada por el programa de Borja",
          completed: true
      }
      this.servicio.putTarea(tarea).subscribe({
      //función a ejecutar si la petición ha ido bien
      next: (todos) => {
                        console.log(todos);
                        this.tarea=todos;
                      },
      //función a ejecutar si hay un error
      error: (error) => console.log('Algo fue mal',error),
      //funcion a ejecutar si la tarea se completa
      complete: () => console.log('Tarea completada')
      })
  }
```

CRUD - ANGULAR

PASO 9 Modificar Tarea
Modificamos todo.component.html añadiendo al final un botón para ejecutar el alta y comprobar que funciona

```
…
<button (click)="modificarTarea();">Modificar Tarea</button>
…
```

CRUD - ANGULAR

PASO 10 Borrar Tarea
Modificamos todo.service.ts añadiendo el método de modificar

```
//método para borrar una tarea
 public deleteTarea(id?:number):Observable<void>{
    //Borramos el todo que y se lo pasamos por la URL
    return this.http.delete<void>(`${this.api}/id}`);
 }
```

CRUD - ANGULAR

PASO 11 Borrar Tarea

Modificamos todo.component.ts añadiendo el método de modificación

```
public borrarTarea() {
    let tarea = this.servicio.deleteTarea(this.id).subscribe({
    //función a ejecutar si la petición ha ido bien
    next: () => {
                    console.log('La tarea '+this.id+' se ha borrado');

                },
    //función a ejecutar si hay un error
    error: (error) => console.log('Algo fue mal',error),
    //funcion a ejecutar si la tarea se completa
    complete: () => console.log('Tarea completada')
    })
 }
```

CRUD - ANGULAR

PASO 11 Borrar Tarea

Modificamos todo.component.ts el constructor para añadir la opción de borrar

```
constructor (private servicio:TodoService,private rutaActiva:ActivatedRoute) {
        this.listaTareas=undefined;
        this.tarea=undefined;
        let url = this.rutaActiva.snapshot.url[0].path;
        if (this.id = this.rutaActiva.snapshot.params['id']) {
            if (url=='borrarTarea') {
              this.borrarTarea();
            } else {
              this.consultaTarea();
            }
        } else {
          this.consultaTareas();
        }
    }
```

CRUD - ANGULAR

PASO 12 Borrar Tarea

Modificamos todo.component.html añadiendo en el listado el borrarTarea

```
<ng-template #listTareas>
<h1>Lista de tareas</h1>
<div>
        <div *ngFor="let t of listaTareas">
                <span class="id" >{{t.id}} </span>
                <span class="tarea">
                        <a [routerLink]="['/borrarTarea',t.id]">
                                Borrar la tarea
                        </a>
                        <a [routerLink]="['/listaTareas',t.id]">
                                {{t.title}}
                        </a>
                </span>
        </div>
</div>
</ng-template>
```

CRUD - ANGULAR

PASO 12 Borrar Tarea

Modificamos app-routing.module.ts para añadir el nuevo path

```
…
 {path:'borrarTarea/:id',component:TodoComponent},
…
```

Angular 18

Podéis realizar el mismo ejemplo utilizando cualquiera de los recursos que hay en https://jsonplaceholder.typicode.com/

Generalitat de Catalunya
Consorci per a la Formació Contínua de Catalunya

pime Coneixement

MINISTERIO DE TRABAJO, MIGRACIONES Y SEGURIDAD SOCIAL
SERVICIO PÚBLICO DE EMPLEO ESTATAL

Angular
IFCD65

Formador: Borja Mulleras Vinzia