

Angular 18 CRUD Example - Sachin Mahore

Before we start, ensure you have the following installed on your system:

1. **Node.js:** [Download and install Node.js](#)
2. **Angular CLI:** Install Angular CLI globally by running the following command:

```
cmd
npm install -g @angular/cli
```

1. Create a New Angular Project

First, create a new Angular project using Angular CLI:

```
VS - Terminal
ng new Angular18SachinM
cd Angular18SachinM
```



Would you like to share pseudonymous usage data about this project with the Angular Team at Google under Google's Privacy Policy at <https://policies.google.com/privacy>. For more details and how to change this setting, see <https://angular.dev/cli/analytics>.

www.eSankalp.com

Options for Responding:

- **Yes:** If you agree to share data, the Angular CLI will send anonymized information about your project and usage to Google. This includes details about Angular versions, commands run, and other non-personal details.

Do you want to enable Server-Side Rendering (SSR) and Static Site Generation (SSG/Prerendering)? (y/N)

- Enable SSR/SSG if:
 - You need SEO optimization for your app, especially if it has dynamic content.

Would you like to use the Server Routing and App Engine APIs (Developer Preview) for this server application?

- You're deploying your Angular Universal app (SSR-enabled) to **Google Cloud App Engine**.

2. Install Bootstrap

```
PS D:\Angular18SachinM> cd Angular18SachinM
```

```
PS D:\Angular18SachinM\Angular18SachinM> npm install bootstrap --save
```

Now after successfully run above command. let's import it in angular.json file.

angular.json

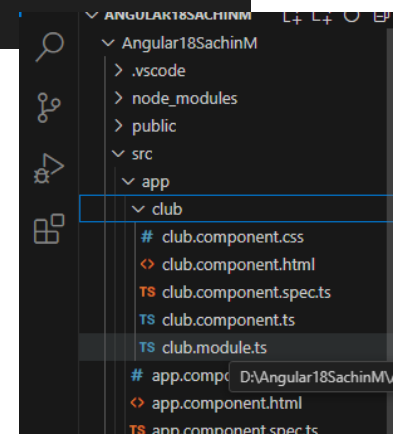
```
"styles": [
  "node_modules/bootstrap/dist/css/bootstrap.min.css",
  "src/styles.css"
],
```

3. Create Club Module

```
ng generate module club
```

run successfully command, it will create files as like bellow path:

src/app/club/club.module.ts



4. Create Component for Module

ng generate component club

it will create folder with files as like bellow path:

src/app/club/

5. Create Routes

In this step, we will simply create route for index, create, edit and view using generated new component. so we have to update our **app.routes.ts** module file as like bellow code:

```
import { ClubComponent } from './club/club.component';
export const routes: Routes = [
  { path: 'club', component: ClubComponent },
];
```

6. Create Interface

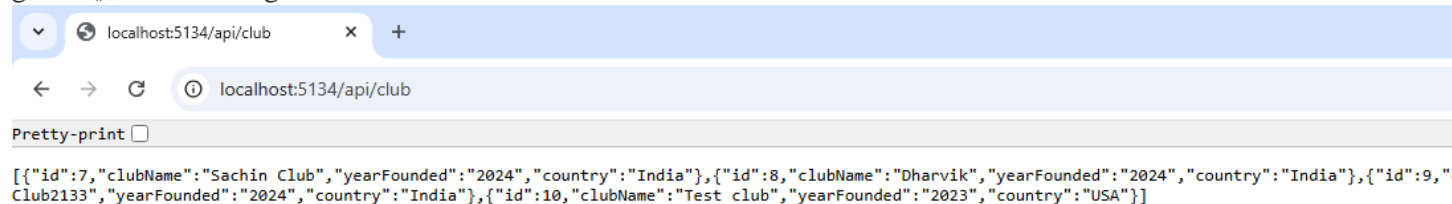
in this step, we will create interface using angular command for club module. we will use club interface with Observable. so let's create interface with bellow code.

ng generate interface club

```
export interface Club {
  id: number;
  clubName: string;
  yearFounded: string;
  country: string;
}
```

7. Create Service (GET)

Here, we will create club service file and we will write and call all web services. we will create getAll(), we are using ASP.NET CORE API



ng generate service club it creates App/Club.Service.ts

```
import { HttpClient, HttpHeaders } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { catchError, throwError } from 'rxjs';
import { Observable } from 'rxjs/internal/Observable';

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

```

private apiUrl = "http://localhost:5134/api";

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)
    )
}

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);
}
}

```

www.eSankalp.com

8. Update Component Logic and Template

Now in this step, we will work on our created component for crud application. now, here we will work on club component. we will call club service and display

src/app/club/club.component.ts

```

import { CommonModule } from '@angular/common';
import { Component } from '@angular/core';
import { RouterModule } from '@angular/router';
import { ClubService } from '../club.service';
import { Club } from '../club';

@Component({
  selector: 'app-club',
  standalone: true,
  imports: [CommonModule, RouterModule],
  templateUrl: './club.component.html',
  styleUrls: ['./club.component.css']
})

```

```
export class ClubComponent {
  clubs : Club[] = [];
  constructor(public clubService : ClubService){}
  ngOnInit(): void {
    this.clubService.getAll().subscribe((data: Club[])=>{
      this.clubs = data;
      console.log(this.clubs);
    }) }}

```



src/app/club/club.component.html

```
<div class="container">
  <h1>Angular 18 CRUD Example - Sachin Mahore</h1>
  <table class="table table-striped">
    <thead>
      <tr>
        <th>ID</th>
        <th>Club Name</th>
        <th>Year Founded</th>
        <th>Country</th>
      </tr>
    </thead>
    <tbody>
      <tr *ngFor="let club of clubs">
        <td>{{ club.id }}</td>
        <td>{{ club.clubName }}</td>
        <td>{{ club.yearFounded }}</td>
        <td>{{ club.country }}</td>
      </tr>
    </tbody>
  </table>
</div>

```

9. Add the logic for Create Club (POST)

Add service for Create club in App/Club.Service.ts

```
create(post:Club): Observable<any> {
  return this.httpClient.post(this.apiUrl + '/posts/', JSON.stringify(post),
    this.httpOptions)
    .pipe(
      catchError(this.errorHandler)
    )
}

```

Update src/app/club/club.component.ts

```
import { ReactiveFormsModule, FormGroup, FormControl, Validators } from
  '@angular/forms';
imports: [CommonModule, RouterModule,ReactiveFormsModule],

```

Add in ngOnInit

```
this.form = new FormGroup({
  clubName: new FormControl('', [Validators.required]),
  yearFounded: new FormControl('', [Validators.required]),
  country: new FormControl('', Validators.required)
});
```

Add in ClubComponent

```
form!: FormGroup;
```

```
submit(){
  console.log(this.form.value);
  this.clubService.create(this.form.value).subscribe((res:any) => {
    console.log(Club created successfully!);
    this.router.navigateByUrl('club');
  })
}
```

src/app/club/club.component.html

```
<h2>Add Club</h2>
<form [formGroup]="form" (ngSubmit)="submit()">
  <div class="form-group">
    <label for="clubName">Club Name:</label>
    <input
      FormControlName="clubName"
      id="clubName"
      type="text"
      class="form-control">
  </div>
  <div class="form-group">
    <label for="country">Country</label>
    <input
      FormControlName="country"
      id="country"
      type="text"
      class="form-control"/>
  </div>
  <div class="form-group">
    <label for="yearFounded">Year Founded</label>
    <input
      FormControlName="yearFounded"
      id="yearFounded"
      type="text"
      class="form-control"/>
  </div>
</form>
```

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

→ localhost:4200/club ☆

Angular 18 CRUD Example - Sachin Mahore

Add Club

Club Name:

Country:

Year Founded:

ID	Club Name	Year Founded	Country
7	Sachin Club	2024	India
8	Dharvik	2024	India
9	Sachin's Club2133	2024	India

10. Add the logic for Get Club Details (GET)

Add service for Get club details in App/Club.Service.ts

```
find(id:number): Observable<any> {
  return this.httpClient.get(this.apiUrl + '/club/' + id)
    .pipe(
      catchError(this.errorHandler)
    )
}
```

Add new Component **ng generate component club/details**

Add in Constructor in club/details/details.component.ts

```
public clubService : ClubService, private route: ActivatedRoute, private router: Router
```

```
ngOnInit(): void {
  this.id = this.route.snapshot.params['clubId'];
  console.log(this.id);
  this.clubService.find(this.id).subscribe((data: Club)=>{
    this.club = data;
  });}
});}
```

Add in details.component.html

```
<div class="container">
<h1>View club</h1>
<a href="/club" routerLink="/club" class="btn btn-primary">Back</a>
```

```

<div>
  <h2>Club Details</h2>
  <div>
    <strong>Title:</strong>
    <p>{{ club.clubName }}</p>
  </div>
  <div>
    <strong>Country:</strong>
    <p>{{ club.country }}</p>
  </div>
  <div>
    <strong>Year Founded:</strong>
    <p>{{ club.yearFounded }}</p>
  </div>
</div>
</div>

```

www.eSankalp.com

Add in **app.routes.ts**

```
{ path: 'club/:clubId/details', component: DetailsComponent
```

Add in **club/club.component.html**

```

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

```

11. Add the logic for Edit / Update Club Details from list (GET)

Add select method in club/club.component.ts . We use same POST service for update

```

this.form = new FormGroup({
  id: new FormControl(0, [Validators.required]),
  clubName: new FormControl('', [Validators.required]),
  yearFounded: new FormControl('', [Validators.required]),
  country: new FormControl('', Validators.required)
});

```

```

selectClub(club: any): void {
  if (club) {
    this.form.patchValue({
      id: club.id,
      clubName: club.clubName || '',
      yearFounded: club.yearFounded || '',
      country: club.country || ''
    });
  }
}

```

update in club/club.component.html in table list

```
<tr *ngFor="let club of clubs" (click)="selectClub(club)">
```

Pass id for Create/Update POST API pass 0 for new record

```
<input type="hidden" FormControlName="id" id="id" value="0" class="form-control">
```

12. Add the logic for Delete Club Details from list (DELETE)

Add in `club/club.component.html`

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

Add service for Delete club details in `App/Club.Service.ts`

```
delete(id:number){
  return this.httpClient.delete(this.apiUrl + '/club/' + id, this.httpOptions)
    .pipe(
      catchError(this.errorHandler)
    )
}
```

Add Delete method in `club/club.component.ts`

```
deletePost(id:number){
  this.clubService.delete(id).subscribe(res => {
    this.clubs = this.clubs.filter(item => item.id !== id);
    console.log('Club deleted successfully!');
  })
}
```

Angular 18 CRUD Example - Sachin Mahore

Add Club

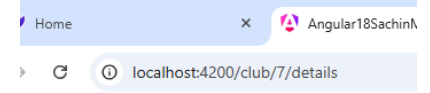
Club Name:

Country:

Year Founded:

ID	Club Name	Year Founded	Country	Action
7	Sachin Club	2024	India	<input type="button" value="View"/> <input type="button" value="Delete"/>
8	Dharvik Mahore	2022	India	<input type="button" value="View"/> <input type="button" value="Delete"/>
9	Sachin's Club 4565	2024	India	<input type="button" value="View"/> <input type="button" value="Delete"/>

www.eSankalp.com



View club

Club Details

ID:

7

Title:

Sachin Club

Country:

India

Year Founded:

2024

Backend ASP.NET CORE API Controllers

```
[Route("api/[controller]")]
[ApiController]
public class ClubController : ControllerBase
{
    private IClubService _clubService;
    public ClubController(IClubService clubService)
    {
        _clubService = clubService;
    }

    [HttpGet]
    public async Task<IActionResult> Get()
    {
        return Ok(await _clubService.GetClubs());
    }
    [HttpPost]
    public async Task<IActionResult> Post(models.Club club)
    {
        if (club.Id == 0)
        {
            await _clubService.AddClub(club);
            return Ok(club.Id);
        }
        else
        {
            await _clubService.UpdateClub(club);
            return Ok(club.Id);
        }
    }

    [HttpGet("{id}")]
    public async Task<IActionResult> GetDetails(int id)
    {
        return Ok(await _clubService.GetClub(id));
    }

    [HttpDelete("{id}")]
    public async Task<IActionResult> Delete(int id)
    {
        await _clubService.DeleteClub(id);
        return Ok();
    }
}
```

www.eSankalp.com