

**VIJAY M**

**Day - 16**

**Date-28/08/25**

**Calling deleteById api to c# from angular**

**calling fetchallnewBook api from angular to c# which returns nested json and display the list**

**source code:**

**my-comp.component.html**

```
<p class="app-title">My Component Works!</p>
```

```
<h3>{{ ooty.Name }} is at {{ ooty.elevation }} m</h3>
```

```
<!-- Favorite Places Section -->
```

```
<section class="form-section">
```

```
  <h2>Add Favorite Place</h2>
```

```
  <input type="text" [(ngModel)]="newPlace" placeholder="Enter your favorite place" />
```

```
  <input type="number" [(ngModel)]="newelevation" placeholder="Enter elevation (in meters)" />
```

```
</section>
```

```
<section class="form-section">
```

```
  <h2>Add a Book</h2>
```

```
  <input type="text" [(ngModel)]="title" placeholder="Enter title" />
```

```
  <input type="text" [(ngModel)]="author" placeholder="Enter author" />
```

```
  <input type="text" [(ngModel)]="genre" placeholder="Enter genre" />
```

```
  <input type="number" [(ngModel)]="price" placeholder="Enter price" />
```

```
  <input type="date" [(ngModel)]="date" placeholder="Enter published date" />
```

```
  <button (click)="addmore()">Add</button>
```

```
</section>
```

```
<section class="form-section">
```

```
  <h2>Delete Book by ID</h2>
```

```

    <input #idInput type="number" placeholder="Enter a book ID" />

<button (click)="deletebyId(+idInput.value)">Delete</button>

</section>

<!-- Favorite Places List -->

<section>

    <h2>My Favorite Places</h2>

    <ul>

        <li *ngFor="let myplace of my_fav_places">

            {{ myplace.Name }} is at {{ myplace.elevation }} m

        </li>

    </ul>

</section>

<!-- Book List -->

<section>

    <h2>Book List</h2>

    <ul>

        <li *ngFor="let book of books">

            <strong>{{ book.title }}</strong> by {{ book.author }}

        </li>

    </ul>

</section>

<h3>New Books</h3>

<ul>

    <li *ngFor="let book of newBooks">

        {{ book.title }} by {{ book.author.authorName }} - ₹{{ book.price }}

    </li>


```

</ul>

### **Model/NewBook.interface.ts**

```
import { Author } from "../Author.interface"

export interface NewBook{

    newBookId:number

    title:string

    price:number

    authorId:string

    author:Author

}
```

### **Model/Author.interface.ts**

```
export interface Author{

    authorId:number

    authorName:string

}
```

### **my-comp.component.ts**

```
import { Component } from '@angular/core';

import {Place} from '../models/place.interface'

import { BookService } from '../book.service';

import { Book } from '../models/book.interface';

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

import { __values } from 'tslib';

import { NewBook } from '../models/NewBook.interface';

@Component({
```

```

    selector: 'app-my-comp',

    templateUrl: './my-comp.component.html',
    styleUrls: ['./my-comp.component.css']
  })
}

export class MyCompComponent {
  constructor(private bookservice:BookService){

  }

  ooty:Place={Name:'tuty',elevation:4}

  newPlace:string=""

  newelevation:number=0

  title:string="";

  author:string="";

  price:number=0;

  date:Date=new Date();

  genre:string="";

  addmore(){

    console.log("button clicked");

    //this.my_fav_places.push({Name:'kanyakumari',elevation:0});

    this.my_fav_places=[...this.my_fav_places,{Name:this.newPlace,elevation:this.newelevation

    }];

    this.bookservice.addbook({

    title:this.title,

    author: this.author,

    price: this.price,

    publishedDate:this.date,

    genre: this.genre
  
```

```

}).subscribe(

  value => {

    console.log(`added: ${value}`);

  },

  error => {

    console.error('Error adding book:', error);

  }

);

this.getbook();  }

my_fav_places:Place[]=this.bookservice.my_fav_places;

books: Book[] = [

// {

//   title: "",

//   author: "",

//   price: 0,

//   id: 0,

//   publishedDate: new Date("2025-08-28T06:57:46.291Z"),

//   genre: ""

// }

];

place_from_service: string = "";

ngOnInit(){

//   this.place_from_service=this.bookservice.getfavplaces();

// console.log(`from servive ${this.place_from_service}`);

console.log("before calling service");

this.getbook();

```

```

        this.FetchAllNewBook();
    }

    ngAfterContentInit(){
        console.log("after content initialized")
    }

    deleteById(id:number){
        this.bookservice.deleteById(id).subscribe(
            value=>{
                console.log(value);
            },
            (error) => {
                this.deleteMessage = `✖ Failed to delete book: ${error.message || 'Unknown error'}`;
                console.error(error);
            }
        )
    }

    getbook(){
        this.bookservice.getbook().subscribe(
            value=>{
                this.books=value;
                console.log(this.books[0].author);
            }
        );
    }

    newBooks:NewBook[]=[];

    FetchAllNewBook(){

```

```

this.bookservice.FetchAllNewBook().subscribe(

  response=>{

    this.newBooks=response;

    console.log(response);

  }

);
}
}

```

### **bookservice.ts**

```

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

import {Place} from './models/place.interface'

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

import { Book } from './models/book.interface';

import { Observable } from 'rxjs';

import { NewBook } from './models/NewBook.interface';

@Injectable({
  providedIn: 'root'
})

export class BookService {

  constructor(private http:HttpClient) { }

  my_fav_places:Place[]=[

    {Name:'trichy',elevation:81},

    {Name:'thanjavur',elevation:88},

    {Name:'thirunelveli',elevation:47}

  ];

```

```
private apiUrl = 'https://localhost:7092/Book';

getbook(): Observable<Book[]> {

    this.http.get<Book[]>(`${this.apiUrl}/GetAllBooks`)

        .subscribe(

            response => {

                // console.log(response); // handle the response here

            },

            error => {

                console.error(error); // handle errors here

            }

        );

    return this.http.get<Book[]>(`${this.apiUrl}/GetAllBooks`);

}

deletebyId(id:number){

    return this.http.delete(`${this.apiUrl}/DeleteBook/${id}`, {

        responseType: 'text'

    });

}

getfavplaces():string{

    return "thainland"

}

addbook(book: Book): Observable<string> {

    const

    token:string="eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWxzb2FwLm9yZy93cy8yMDA1LzA1L2lkZW50aXR5L2NsYWltcy9uYWw1IjoiYWVWYWRtaW4iLCJYXNzd29yZCI6IjU5OTQ0NzFhYmIwMTExMmFmY2MxODE1OWY2Y2M3NGI0ZjUxMWI5OTgwNmRhNTliM2NhZjVhOWMxNzNjYWNNmYzUiLCJleHAiOiE3NTYzNzUzNTQsImIzcyI6Imh0dHBzOi8vd3d3LnBheW9kYS5jb20iLCJhdWQiOiJtdHVkZW50c19DU2hhcnAifQ.5qV9Le_hUvzyjvnZViMnp0tG351V2952ABIFYnu3cgs";
```



```
const header=new
HttpHeaders().set('Content-Type','application/json').append('Authorization','Bearer ${token}');

return this.http.post(`${this.apiUrl}/AddBook`, book, {

    headers:header,

    responseType: 'text'

});

}

FetchAllNewBook():Observable<NewBook[]>{

    return this.http.get<NewBook[]>(`${this.apiUrl}/FetchAllNewBook`)

}

}
```

**output:**

⌕

⌕

Elements

Console

Sources

Network

Performance

Memory

Application

>>

⌕

⌕

top

⌕

Filter

Default levels

8 Issu

[webpack-dev-server] Server started: Hot Module Replacement disabled, Live Reloading enabled, Progress disabled, Overlay enabled.

before calling service

after content initialized

Angular is running in development mode. Call enableProdMode() to enable production mode.

rowling

(10) [{...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}]

0:

author: {authorId: 1, authorName: 'rowling'}

authorId: 1

newBookId: 1

price: 1000

title: "harry"

[[Prototype]]: Object

1: {newBookId: 2, title: 'agnisiragu', price: 0, authorId: 2, author: {...}}

2: {newBookId: 3, title: 'how to win friends', price: 400, authorId: 3, author: {...}}

3: {newBookId: 4, title: 'siragu', price: 100, authorId: 2, author: {...}}

4: {newBookId: 5, title: 'how to win friends', price: 500, authorId: 2, author: {...}}

5: {newBookId: 6, title: 'how friends', price: 500, authorId: 2, author: {...}}

6: {newBookId: 7, title: 'how friends', price: 500, authorId: 2, author: {...}}

7: {newBookId: 8, title: 'how friends', price: 500, authorId: 2, author: {...}}

8: {newBookId: 9, title: 'how friends', price: 500, authorId: 2, author: {...}}

9: {newBookId: 10, title: 'how friends', price: 500, authorId: 2, author: {...}}

length: 10

[[Prototype]]: Array(0)

>

## Add Favorite Place

## Add a Book



## Delete Book by ID

Delete

## My Favorite Places

trichy is at 81 m

thanjavur is at 88 m

thirunelveli is at 47 m

## Delete Book by ID

Delete

✗ Failed to delete book: Http failure response for https://localhost:7092/Book/DeleteBook/13: 404 OK

## Delete Book by ID

Delete

Book deleted successfully

## NEW BOOKS

### New Books

harry by rowling - ₹1000

agnisiragu by kalam - ₹0

how to win friends by yuvaraj - ₹400

siragu by kalam - ₹100

how to win friends by kalam - ₹500

how friends by kalam - ₹500

how friends by kalam - ₹500

how friends by kalam - ₹500

how friends by kalam - ₹500

how friends by kalam - ₹500

