VIJAY M

Day -15

Date-26/08/25

From typescript, call the api to delete book via API

-after the deletion, call the fuction to fetch the items and using foreach iterate then display it.

Souce code

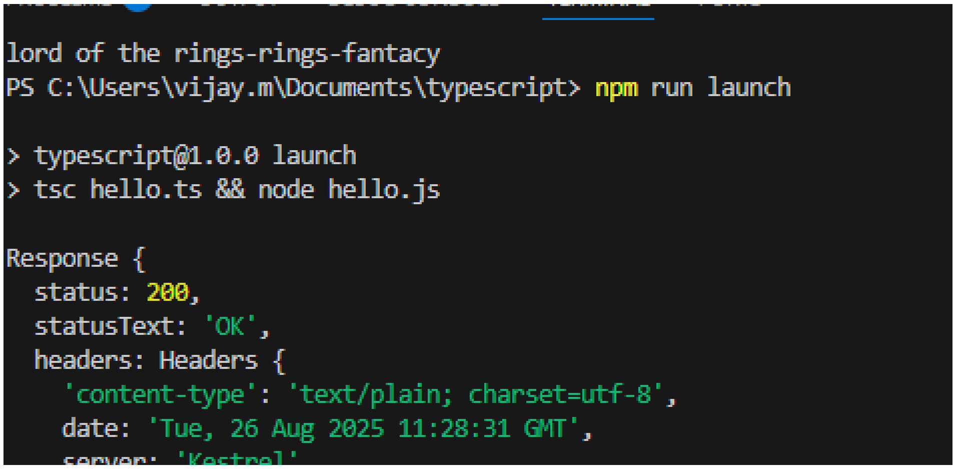
index.ts

```
// let txt:string ="hello world";
// let age:number=18;
// console.log(typeof(age));
// var edu:string="school";
// let edu_let:string="school";
// console.log(edu);
// console.log("--"+edu_let);
// if(age>17){
//          var edu="college";
//          let edu_let="college";
//          console.log(edu);
//          console.log("--"+edu_let);
// }
// console.log(edu);
// console.log("--"+edu_let);
// console.log(`name:${txt},age:${age}`);
// let tuple:[string,number]=["martin",44];
// console.log(`tuple is ${tuple} of type ${typeof(tuple)}`);

// enum Color {red,green,blue};

// let c:Color=Color.green;

// console.log(c);

// function setpersondetails(txt:string,age:number=23):string{

//          console.log(`name:${txt},age:${age}`);

//          return "hii";

// }

// console.log(setpersondetails("kavay"));

// setpersondetails("nandhada leo",50);

// type options="rick"|34|true;

// let opt_type:options=34;

// opt_type=true;

// let details=function(){

//          console.log("i am sam");

//          return "how are u";

// }

// details();

// let details_lamda=()=>console.log("moon night");

// details_lamda();

// interface IPerson{

//          nam:string

//          age:number

// }

// interface IMovie{

//          title:string
```

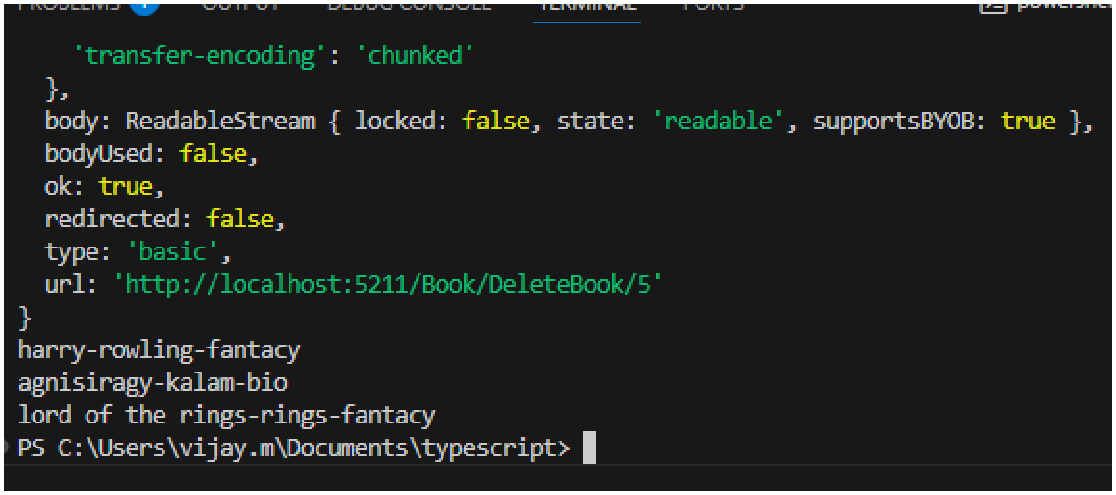
```
//	rating:number
// }
// interface IDirector extends IPerson,IMovie{
//	quality:string
// }
// let james:IPerson={
//	nam:"james cameron",
//	age:80
// }
// let avator:IMovie={
//	title:"avator fire and ash",
//	rating:8.1
// }
// let nolan:IDirector={
//	nam:"nolan",
//	age:65,
//	title:"inception",
//	rating:9.0,
//	quality:"imax"
// }
// console.log(`director:${nolan.title}`);
// class Car{
//	model:string;
//	company:string;
//	cost:number
//	constructor(model:string,company:string,cost:number){
```

```
//                  this.model=model;
//                  this.company=company;
//                  this.cost=cost;
//          }
// }
// let bmw:Car=new Car("t1","bmw",9000000);
// console.log(`Car:${bmw.cost}`);
interface bookDetails{
        id:number,
        title:string,
        author:string,
        publishedDate:Date,
        genre:string,
        price:number
}
interface book_det{
        authorName:string;
        bookTitle:string;
        price:number;
}
interface user{
        Username:string;
        Password:string;
}
const usr:user={
        Username:"admin",
    Password:"12345",
}
const book:book_det={
    bookTitle:"how friends",
    authorName:"kalam",
    price:500
}
async function addnewbokk(token:string) {
    const response=await fetch(`http://localhost:5211/Book/AddNewBook`,{method:'POST',
        body:JSON.stringify(book),
        headers:{
            'Content-Type':'application/json',
            'Authorization':'Bearer '+token
        }
    });
    if(!response.ok){
        console.log("error occured");
    }
    console.log(response);
}
async function login(callback_func:(token:string)=>(void)){
    const response=await fetch(`http://localhost:5211/User/Login`,{method:'POST',
        body:JSON.stringify(usr),
        headers:{
            'Content-Type':'application/json',
        }
    });

    if(!response.ok){

        console.log("error occured");

    }

    const token=await response.text();

    console.log(token);

    //callback_func(token);

}

//login(addnewbokk);

// async function main() {

//      await login();

//      if (tokens) {

//          await addnewbokk(tokens);

//      } else {

//          console.log("No token received; cannot add book.");

//      }

// }

// main();

async function fetchBooks() {

    const response=await fetch(`http://localhost:5211/Book/GetAllBooks`,{

        method:'POST'

    });

    if(!response.ok){

        console.log("error occured");

    }

    // const res:bookDetails[]=await response.json();
    // for(let i=0;i<res.length;i++){
    //          console.log(res[i].title);
    // }
    const res:[bookDetails]=await response.json();
    res.forEach(
        (thisbook:bookDetails)=>{
            console.log(`${thisbook.title}-${thisbook.author}-${thisbook.genre}`);
        }
    )


}
//fetchBooks();
async function DeleteBookById(id:number) {
    const response =await fetch(`http://localhost:5211/Book/DeleteBook/${id}`,{
        method:'DELETE'
    });
    if(!response.ok){
        console.log("error occured");
    }
    console.log(response);
    fetchBooks();
}
DeleteBookById(5);
```

Output



```
lord of the rings-rings-fantacy
PS C:\Users\vijay.m\Documents\typescript> npm run launch

> typescript@1.0.0 launch
> tsc hello.ts && node hello.js

Response {
  status: 200,
  statusText: 'OK',
  headers: Headers {
    'content-type': 'text/plain; charset=utf-8',
    date: 'Tue, 26 Aug 2025 11:28:31 GMT',
    server: 'Kestrel',
```



```
    'transfer-encoding': 'chunked'
  },
  body: ReadableStream { locked: false, state: 'readable', supportsBYOB: true },
  bodyUsed: false,
  ok: true,
  redirected: false,
  type: 'basic',
  url: 'http://localhost:5211/Book/DeleteBook/5'
}
harry-rowling-fantacy
agnisiragy-kalam-bio
lord of the rings-rings-fantacy
PS C:\Users\vijay.m\Documents\typescript>
```