Angular TS – API Integration

[**✅ Task 8: HTTP Client – API Integration (GET, POST, PUT, DELETE)** 1](#_Toc205368233)

[**1️⃣ Purpose** 1](#_Toc205368234)

[**2️⃣ Theory** 1](#_Toc205368235)

[**3️⃣ Prerequisites** 1](#_Toc205368236)

[**4️⃣ Code Example (Step-by-Step)** 2](#_Toc205368237)

[**5️⃣ Project Structure Snapshot** 4](#_Toc205368238)

[**6️⃣ Summary** 4](#_Toc205368239)

**✅ Task 8: HTTP Client – API Integration (GET, POST, PUT, DELETE)**

**1️⃣ Purpose**

To enable communication with a backend Web API using Angular’s HttpClient for performing **CRUD operations**:

* Fetch data (GET)
* Create data (POST)
* Update data (PUT)
* Delete data (DELETE)

**2️⃣ Theory**

| **Concept** | **Description** |
| --- | --- |
| HttpClient | Service from @angular/common/http to perform HTTP requests |
| Observable | Returned from HttpClient methods to handle async response |
| subscribe() | Used to trigger the HTTP request and handle the response |
| HttpClientModule | Must be imported for using HttpClient |

**3️⃣ Prerequisites**

✅ Backend API (e.g., a .NET Web API or mock API like JSONPlaceholder)  
✅ Import HttpClientModule in main.ts or your app.config.ts

**4️⃣ Code Example (Step-by-Step)**

Let’s create a StudentService and use it in a StudentApiComponent to perform CRUD operations.

**✅ a) Add HttpClientModule to App Config**

// src/app/app.config.ts

import { provideHttpClient } from '@angular/common/http';

export const appConfig: ApplicationConfig = {

providers: [

provideRouter(appRoutes),

provideHttpClient() // ✅ Required for HttpClient

]

};

**✅ b) Create StudentService**

ng generate service services/studentapis

// src/app/services/student.service.ts

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

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

import { Observable } from 'rxjs';

export interface Student {

id: number;

name: string;

email: string;

}

@Injectable({

providedIn: 'root'

})

export class StudentapiService {

private apiUrl = 'https://jsonplaceholder.typicode.com/users'; // Replace with your API URL

constructor(private http: HttpClient) {}

getStudents(): Observable<Student[]> {

return this.http.get<Student[]>(this.apiUrl);

}

getStudent(id: number): Observable<Student> {

return this.http.get<Student>(`${this.apiUrl}/${id}`);

}

createStudent(student: Student): Observable<Student> {

return this.http.post<Student>(this.apiUrl, student);

}

updateStudent(student: Student): Observable<Student> {

return this.http.put<Student>(`${this.apiUrl}/${student.id}`, student);

}

deleteStudent(id: number): Observable<any> {

return this.http.delete(`${this.apiUrl}/${id}`);

}

}

**✅ c) Create API Component**

ng generate component components/student-api --standalone

**✅ d) student-api.component.ts**

import { Component, OnInit } from '@angular/core';

import { CommonModule } from '@angular/common';

import { StudentService, Student } from '../../services/student.service';

import { ReactiveFormsModule, FormGroup, FormControl } from '@angular/forms';

@Component({

selector: 'app-student-api',

standalone: true,

imports: [CommonModule, ReactiveFormsModule],

template: `

<h2>📡 Student API (CRUD)</h2>

<form [formGroup]="studentForm" (ngSubmit)="onSubmit()">

<input formControlName="name" placeholder="Name" />

<input formControlName="email" placeholder="Email" />

<button type="submit">Add Student</button>

</form>

<ul>

<li \*ngFor="let student of students">

{{ student.name }} - {{ student.email }}

<button (click)="deleteStudent(student.id)">❌ Delete</button>

</li>

</ul>

`

})

export class StudentApiComponent implements OnInit {

students: Student[] = [];

studentForm = new FormGroup({

name: new FormControl(''),

email: new FormControl('')

});

constructor(private studentService: StudentService) {}

ngOnInit() {

this.loadStudents();

}

loadStudents() {

this.studentService.getStudents().subscribe(data => {

this.students = data;

});

}

onSubmit() {

const student = this.studentForm.value as Student;

this.studentService.createStudent(student).subscribe(res => {

this.loadStudents(); // Refresh list

this.studentForm.reset();

});

}

deleteStudent(id: number) {

this.studentService.deleteStudent(id).subscribe(() => {

this.loadStudents();

});

}

}

**✅ e) Add Route in app.routes.ts**

import { StudentApiComponent } from './components/student-api/student-api.component';

export const appRoutes: Routes = [

{ path: 'student-api', component: StudentApiComponent },

// other routes...

];

**5️⃣ Project Structure Snapshot**

src/

├── app/

│ ├── services/

│ │ └── studentapis.service.ts ✅

│ ├── components/

│ │ └── student-api/ ✅

│ ├── app.config.ts

│ └── app.routes.ts

**6️⃣ Summary**

✅ You now know how to:

* Use HttpClient for GET, POST, PUT, DELETE.
* Create a reusable service (StudentapiService).
* Use the service in a standalone component (StudentApiComponent).
* Bind data using reactive forms.