Ejemplos de Código en Angular para consumir una API REST

# 5. Configurar HttpClientModule en app.module.ts

import { HttpClientModule } from '@angular/common/http';  
  
@NgModule({  
 declarations: [...],  
 imports: [  
 ...,  
 HttpClientModule  
 ],  
 providers: [],  
 bootstrap: [AppComponent]  
})  
export class AppModule { }

# 6. Implementar métodos en api.service.ts

import { Injectable } from '@angular/core';  
import { HttpClient } from '@angular/common/http';  
import { Observable } from 'rxjs';  
  
@Injectable({  
 providedIn: 'root'  
})  
export class ApiService {  
  
 private baseUrl = 'https://jsonplaceholder.typicode.com/users';  
  
 constructor(private http: HttpClient) {}  
  
 // GET - obtener todos  
 getUsers(): Observable<any[]> {  
 return this.http.get<any[]>(this.baseUrl);  
 }  
  
 // GET - obtener uno por ID  
 getUserById(id: number): Observable<any> {  
 return this.http.get<any>(`${this.baseUrl}/${id}`);  
 }  
  
 // POST - crear usuario  
 createUser(user: any): Observable<any> {  
 return this.http.post<any>(this.baseUrl, user);  
 }  
  
 // PUT - actualizar usuario  
 updateUser(id: number, user: any): Observable<any> {  
 return this.http.put<any>(`${this.baseUrl}/${id}`, user);  
 }  
  
 // DELETE - eliminar usuario  
 deleteUser(id: number): Observable<any> {  
 return this.http.delete<any>(`${this.baseUrl}/${id}`);  
 }  
}

# 7. Usar el servicio en app.component.ts

import { Component, OnInit } from '@angular/core';  
import { ApiService } from './services/api.service';  
  
@Component({  
 selector: 'app-root',  
 template: `  
 <h1>Usuarios</h1>  
 <button (click)="loadUsers()">Cargar Usuarios</button>  
 <ul>  
 <li \*ngFor="let user of users">  
 {{ user.name }}   
 <button (click)="deleteUser(user.id)">Eliminar</button>  
 </li>  
 </ul>  
 <button (click)="addUser()">Agregar Usuario</button>  
 `  
})  
export class AppComponent implements OnInit {  
 users: any[] = [];  
  
 constructor(private api: ApiService) {}  
  
 ngOnInit() {  
 this.loadUsers();  
 }  
  
 loadUsers() {  
 this.api.getUsers().subscribe(data => {  
 this.users = data;  
 });  
 }  
  
 addUser() {  
 const newUser = { name: 'Nuevo Usuario', email: 'test@example.com' };  
 this.api.createUser(newUser).subscribe(res => {  
 console.log('Usuario creado:', res);  
 this.loadUsers();  
 });  
 }  
  
 deleteUser(id: number) {  
 this.api.deleteUser(id).subscribe(() => {  
 console.log('Usuario eliminado');  
 this.loadUsers();  
 });  
 }  
}