Http Client -CRUD Demo Steps

# Angular 16 HttpClient & Http Services Example Tutorial

In Angular, HttpClient is an API that allows to perform HTTP requests. HTTP requests are made to create communication between the client and server. The HttpClient service is available as an injectable class and comes with various methods (GET, POST, PUT, UPDATE, DELETE, etc.) to send HTTP requests in Angular.

we will walk you through how to send HTTP requests in an Angular application using the HttpClient and HttpClientModule services. To understand how to make HTTP requests in Angular, we will create a basic employee management system using the Angular framework.

Furthermore, we will cover the following topics, considering the HttpClient API:

* How to make an HTTP GET request in Angular
* How to send an HTTP POST request in Angular
* How to send an HTTP PUT request to update data on a server in Angular
* How to make an HTTP DELETE request in Angular to delete data or resources on a server

How to set up the HttpClientModule in Angular app, how to set up and use a local server with JSON server NPM package, and how to make GET, POST, PUT & DELETE request with Angular using HttpClient API.

ng new http-demo --no-standalone

cd http-demo

code .

npm install bootstrap jquery –save

angular.json

 "styles": [

              "src/styles.scss",

              "node\_modules/bootstrap/dist/css/bootstrap.min.css"

            ],

ng g c employee-list

ng g c create-employee

ng g c edit-employee

**Configure JSON Server in Angular**

**npm install -g json-server**

**db.json**

{

    "employees": [{

      "id": 1,

      "name": "Geetha",

      "email": "geetha@mail.com",

      "phone": "001-123-4567"

    }, {

      "id": 2,

      "name": "Fransy",

      "email": "fransy@mail.com",

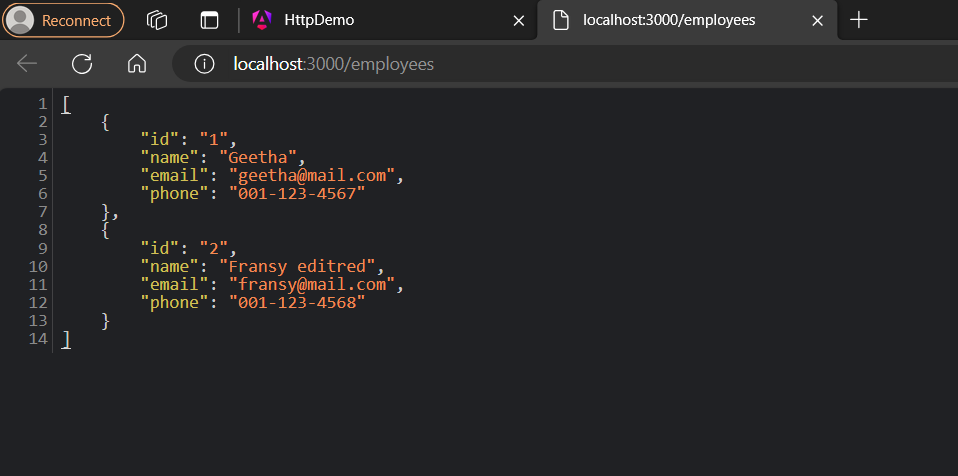
      "phone": "001-123-4568"

    }]

  }

D:\Zuci System Jan 24\Angular MAterial demos\http-demo\src\app>json-server --watch db.json

http://localhost:3000/employees.



**Enable Routing Service in Angular**

App-routing.module.ts

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

import { RouterModule, Routes } from '@angular/router';

import { CreateEmployeeComponent } from './create-employee/create-employee.component';

import { EmployeeListComponent } from './employee-list/employee-list.component';

import { EditEmployeeComponent } from './edit-employee/edit-employee.component';

const routes: Routes = [

   { path: '', pathMatch: 'full', redirectTo: 'employees-list' },

  { path: 'create-employee', component: CreateEmployeeComponent },

  { path: 'employees-list', component: EmployeeListComponent },

  { path: 'employee-edit/:id', component: EditEmployeeComponent },

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

app.component.html

<router-outlet></router-outlet>

**app.module.ts**

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

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

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

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

import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';

import { EmployeeListComponent } from './employee-list/employee-list.component';

import { CreateEmployeeComponent } from './create-employee/create-employee.component';

import { EditEmployeeComponent } from './edit-employee/edit-employee.component';

@NgModule({

  declarations: [

    AppComponent,

    EmployeeListComponent,

    CreateEmployeeComponent,

    EditEmployeeComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule,

    HttpClientModule,

    FormsModule,

    ReactiveFormsModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

**Create Angular Service**

In order to create consume RESTful API using Angular HttpClient service we need to create a service file in our app. This file will hold the core logic of our demo application.

**Functionalities to be covered:**

* Create Employee
* Delete Employee
* Update Employee
* Manage Employee List

In order to create CRUD operations using RESTful API in Angular, we need to generate employee.ts class and rest-api.service.ts files.

Next, generate employee interface class:

ng g i shared/Employee

ng g i shared/Employee

export interface Employee {

    id: string;

   name: string;

   email: string;

   phone: number;

}

Next, generate RestApiService class:

Ng g s shared/rest-api

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

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

import { Employee } from '../shared/employee';

import { Observable, throwError } from 'rxjs';

import { retry, catchError } from 'rxjs/operators';

@Injectable({

  providedIn: 'root',

})

export class RestApiService {

  // Define API

  apiURL = 'http://localhost:3000';

  constructor(private http: HttpClient) {}

  /\*========================================

    CRUD Methods for consuming RESTful API

  =========================================\*/

  // Http Options

  httpOptions = {

    headers: new HttpHeaders({

      'Content-Type': 'application/json',

    }),

  };

  // HttpClient API get() method => Fetch employees list

  getEmployees(): Observable<Employee> {

    return this.http

      .get<Employee>(this.apiURL + '/employees')

      .pipe(retry(1), catchError(this.handleError));

  }

  // HttpClient API get() method => Fetch employee

  getEmployee(id: any): Observable<Employee> {

    return this.http

      .get<Employee>(this.apiURL + '/employees/' + id)

      .pipe(retry(1), catchError(this.handleError));

  }

  // HttpClient API post() method => Create employee

  createEmployee(employee: any): Observable<Employee> {

    return this.http

      .post<Employee>(

        this.apiURL + '/employees',

        JSON.stringify(employee),

        this.httpOptions

      )

      .pipe(retry(1), catchError(this.handleError));

  }

  // HttpClient API put() method => Update employee

  updateEmployee(id: any, employee: any): Observable<Employee> {

    return this.http

      .put<Employee>(

        this.apiURL + '/employees/' + id,

        JSON.stringify(employee),

        this.httpOptions

      )

      .pipe(retry(1), catchError(this.handleError));

  }

  // HttpClient API delete() method => Delete employee

  deleteEmployee(id: any) {

    return this.http

      .delete<Employee>(this.apiURL + '/employees/' + id, this.httpOptions)

      .pipe(retry(1), catchError(this.handleError));

  }

  // Error handling

  handleError(error: any) {

    let errorMessage = '';

    if (error.error instanceof ErrorEvent) {

      // Get client-side error

      errorMessage = error.error.message;

    } else {

      // Get server-side error

      errorMessage = `Error Code: ${error.status}\nMessage: ${error.message}`;

    }

    window.alert(errorMessage);

    return throwError(() => {

      return errorMessage;

    });

  }

}

**Create Data using Angular HTTP POST Request**

employee-create.component.html

<div class="container custom-container mb-5">

    <div class="col-md-12">

      <h3 class="mb-3 text-center">Create Employee</h3>

      <div class="mb-3 ">

        <input type="text" [(ngModel)]="employeeDetails.name" class="form-control" placeholder="Name">

      </div>

      <div class="mb-3 ">

        <input type="text" [(ngModel)]="employeeDetails.email" class="form-control" placeholder="Email">

      </div>

      <div class="mb-3 ">

        <input type="text" [(ngModel)]="employeeDetails.phone" class="form-control" placeholder="Phone">

      </div>

      <div class="form-group">

        <button class="btn btn-success btn-lg btn-block" (click)="addEmployee(employeeDetails)">Create Employee</button>

      </div>

    </div>

  </div>

employee-create.component.ts

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

import { RestApiService } from '../shared/rest-api.service';

import { Router } from '@angular/router';

@Component({

  selector: 'app-create-employee',

  templateUrl: './create-employee.component.html',

  styleUrl: './create-employee.component.scss'

})

export class CreateEmployeeComponent {

  @Input() employeeDetails = { name: '', email: '', phone: 0 };

  constructor(public restApi: RestApiService, public router: Router) {}

  ngOnInit() {}

  addEmployee(dataEmployee: any) {

    this.restApi.createEmployee(this.employeeDetails).subscribe((data: {}) => {

      this.router.navigate(['/employees-list']);

    });

  }

}

**Send HTTP GET and DELETE Requests**

employee-list.component.html

<div class="container custom-container-2 mt-5">

    <!-- Show it when there is no employee -->

    <div class="no-data text-center" \*ngIf="Employee.length == 0">

      <p>There is no employee added yet!</p>

      <button class="btn btn-outline-primary" routerLink="/create-employee">

        Add Empoyee

      </button>

    </div>

    <!-- Employees list table, it hides when there is no employee -->

    <div \*ngIf="Employee.length !== 0">

      <h3 class="mb-3 text-center">Employees List</h3>

      <div class="col-md-12">

        <table class="table table-bordered">

          <thead>

            <tr>

              <th scope="col">User Id</th>

              <th scope="col">Name</th>

              <th scope="col">Email</th>

              <th scope="col">Phone</th>

              <th scope="col">Action</th>

            </tr>

          </thead>

          <tbody>

            <tr \*ngFor="let employee of Employee">

              <td>{{ employee.id }}</td>

              <td>{{ employee.name }}</td>

              <td>{{ employee.email }}</td>

              <td>{{ employee.phone }}</td>

              <td>

                <span

                  class="edit btn btn-sm btn-outline-primary me-2"

                  routerLink="/employee-edit/{{ employee.id }}"

                  >Edit</span

                >

                <span

                  class="delete edit btn btn-sm btn-outline-danger"

                  (click)="deleteEmployee(employee.id)"

                  >Delete</span

                >

              </td>

            </tr>

          </tbody>

        </table>

      </div>

    </div>

  </div>

employee-list.component.ts

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

import { RestApiService } from '../shared/rest-api.service';

@Component({

  selector: 'app-employee-list',

  templateUrl: './employee-list.component.html',

  styleUrl: './employee-list.component.scss'

})

export class EmployeeListComponent {

  Employee: any = [];

  constructor(public restApi: RestApiService) {}

  ngOnInit() {

    this.loadEmployees();

  }

  // Get employees list

  loadEmployees() {

    return this.restApi.getEmployees().subscribe((data: {}) => {

      this.Employee = data;

    });

  }

  // Delete employee

  deleteEmployee(id: any) {

    if (window.confirm('Are you sure, you want to delete?')) {

      this.restApi.deleteEmployee(id).subscribe((data) => {

        this.loadEmployees();

      });

    }

  }

}

edit-employee.component.html

<div class="container custom-container">

    <div class="col-md-12">

      <h3 class="mb-3 text-center">Update Employee</h3>

      <div class="mb-3">

        <input

          type="text"

          [(ngModel)]="employeeData.name"

          class="form-control"

          placeholder="Name"

        />

      </div>

      <div class="mb-3">

        <input

          type="text"

          [(ngModel)]="employeeData.email"

          class="form-control"

          placeholder="Email"

        />

      </div>

      <div class="mb-3">

        <input

          type="text"

          [(ngModel)]="employeeData.phone"

          class="form-control"

          placeholder="Phone"

        />

      </div>

      <div class="form-group">

        <button

          class="btn btn-success btn-lg btn-block"

          (click)="updateEmployee()"

        >

          Update Employee

        </button>

      </div>

    </div>

  </div>

edit-employee.component.ts

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

import { RestApiService } from '../shared/rest-api.service';

import { Router,ActivatedRoute } from '@angular/router';

@Component({

  selector: 'app-edit-employee',

  templateUrl: './edit-employee.component.html',

  styleUrl: './edit-employee.component.scss'

})

export class EditEmployeeComponent {

  id = this.actRoute.snapshot.params['id'];

  employeeData: any = {};

  constructor(

    public restApi: RestApiService,

    public actRoute: ActivatedRoute,

    public router: Router

  ) {

  }

  ngOnInit() {

    this.restApi.getEmployee(this.id).subscribe((data: {}) => {

      this.employeeData = data;

    })

  }

  // Update employee data

  updateEmployee() {

    if(window.confirm('Are you sure, you want to update?')){

      this.restApi.updateEmployee(this.id, this.employeeData).subscribe(data => {

        this.router.navigate(['/employees-list'])

      })

    }

  }

}