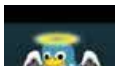
positronX.io

Home » Angular » Angular 15 HttpClient & Htt

Angular 15 HttpClient & Tutorial

Last updated on: **April 24, 2023** by Digamber

FEATURED VIDEOS



Today, we are going to show you how you can consume RESTful API in Angular 12 using HttpClient service. HttpClient service is a very useful API in Angular to communicate with the remote server. In this post, we will teach you how to make HTTP request in Angular.

positronX.io

Angular HttpClient Features

- Observable Support
- Hassle Free API Testing
- Smooth Requests & Response
- Better Error Handling

HttpClient is an injectable service, it comes with the various powerful methods to communicate with the remote server. HttpClient API can send Http POST, GET, PUT and DELETE requests easily.

Angular 15 HttpClient Methods

- `request()`
- `delete()`
- `get()`
- `head()`
- `jsonp()`
- `options()`
- `patch()`
- `post()`
- `put()`

I'll be showing you the practical examples of standard HTTP methods like GET, PUT, POST and DELETE, these methods allow you to communicate with a REST API server.

positronX.io

By the end of this tutorial, you'll be
HttpClientModule in Angular app?
JSON server NPM package, and ho
request with Angular using HttpCli

Angular HttpClient Services Example

- Install Angular CLI
- Configure Fake JSON Server in Angular
- Enable Routing Service in Angular
- Configure Angular HttpClient
- Create Angular Service for Consuming RESTful API using Angular HttpClient API
- Access HttpClient API from Angular Component
- Send HTTP GET and DELETE Requests in Angular to Manage Data
- Make HTTP PUT Request in Angular to Update Data

Create Angular Project

In order to create this demo app you must have **Node JS development environment** set up in your machine.

already installed.

```
npm install @angular/cli -g
```

I will be creating an employee record management system with Angular, in this demo app i will consume RESTful API via HttpClient service.

It's time to setup Angular project, run the following command in Angular CLI.

```
ng new angular-httpclient-app
```

positronX.io

Select y and Hit Enter.

Which stylesheet format would yo

Choose CSS and hit Enter

After that your project will start cre
forget to get into the project's folde

```
cd angular-httpclient-app
```

I'll be using Bootstrap 4 CSS framework with Angular for consuming RESTful API with HttpClient service. Hit the following command to get the Bootstrap in your Angular app.

```
npm install bootstrap
```

After that, Go to `angular.json` file and replace the given below code with "styles": [ ] array.

```
"styles": [
            "node_modules/bootstrap/dist/css/bootstrap.min.css",
            "src/styles.css"
          ]
```

Now, we have to generate the following components.

```
ng g c employee-create
ng g c employee-edit
ng g c employee-list
```

positronX.io

taking help of json-server NPM pac

Install JSON server in our project, ru

```
npm i json-server --save
```

Then, create a folder by the name of **server** and keep your **database** file in it to manage the APIs locally.

```
mkdir server && cd server
```

```
touch db.json
```

Once the `db.json` file is created then add some data in it.

```
{
  "employees": [{
    "id": 1,
    "name": "Tony Stark",
    "email": "tony@mcu.com",
    "phone": "001-123-4567"
  }, {
    "id": 2,
    "name": "Black Widow",
    "email": "black-widow@mcu.com",
    "phone": "001-123-4568"
  }]
}
```

positronX.io

Now if you make any request with A
db.json file will get updated locally.

You can check your local db.json fil

Enable Routing Service in Angula

For navigating between components in Angular we need to activate routing service in our application, to enable routes go to `app-routing.module.ts` file and add the following code.

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';
import { EmployeeCreateComponent } from './employee-create/employee-creat
import { EmployeeEditComponent } from './employee-edit/employee-edit.comp
import { EmployeeListComponent } from './employee-list/employee-list.comp
const routes: Routes = [
  { path: '', pathMatch: 'full', redirectTo: 'create-employee' },
  { path: 'create-employee', component: EmployeeCreateComponent },
  { path: 'employees-list', component: EmployeeListComponent },
  { path: 'employee-edit/:id', component: EmployeeEditComponent },
];
@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule],
})
export class AppRoutingModule {}
```

Enable the routes within view, add the following code in app.component.html file.

```
<router-outlet></router-outlet>
```

positronX.io

Import HttpClient API

In this tutorial, I will give you the de
data using the **RESTful API in Angu**
HttpClient API to make the commu
set up this service in your Angular a

Go to `app.module.ts` and paste the following code.

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

Include the HttpClientModule in `@NgModule's` imports array.

```
@NgModule({
  imports: [
    HttpClientModule
   ]
})
```

You've injected the **HttpClientModule** in your application, now you can use it easily in Angular app.

Besides, here is the complete **app.module.ts** file, which contains routing, forms, app components and http modules.

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppComponent } from './app.component';
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
import { AppRoutingModule } from './app-routing.module';
import { EmployeeCreateComponent } from './employee-create/employee-creat
```

```
    EmployeeCreateComponent,
    EmployeeEditComponent,
    EmployeeListComponent,
  ],
  imports: [
    BrowserModule,
    HttpClientModule,
    FormsModule,
    ReactiveFormsModule,
    AppRoutingModule,
  ],
  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:

- `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
```

positronX.io

```
export class Employee {
   id: string;
   name: string;
   email: string;
   phone: number;
}
```

Next, generate RestApiService class:

```
ng g s shared/rest-api
```

I will be writing down core logic in this file for consuming RESTful API using HttpClient API. We will also use RxJS to handle asynchronous operations and errors in this demo app.

Let's go to `shared/rest-api.service.ts` file and add the following code.

```
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 = {
```

positronX.io

```
    return this.http
      .get<Employee>(this.apiUR
      .pipe(retry(1), catchErro
  }
  // HttpClient API get() metho
  getEmployee(id: any): Observa
    return this.http
      .get<Employee>(this.apiUR
      .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.httpOption
      .pipe(retry(1), catchError(this.handleError));
  }
  // Error handling
  handleError(error: any) {
    let errorMessage = '';
    if (error.error instanceof ErrorEvent) {
      // Get client-side error
```

positronX.io

```
      window.alert(errorMessage);
      return throwError(() => {
        return errorMessage;
      });
    }
  }
```

Create Data using Angular HTTP POST Request

Go to `employee-create.component.html` add the following code.

```
<div class="container custom-container">
  <div class="col-md-12">
    <h3 class="mb-3 text-center">Create Employee</h3>
    <div class="form-group">
      <input type="text" [(ngModel)]="employeeDetails.name" class="form-c
    </div>
    <div class="form-group">
      <input type="text" [(ngModel)]="employeeDetails.email" class="form-
    </div>
    <div class="form-group">
      <input type="text" [(ngModel)]="employeeDetails.phone" class="form-
    </div>
    <div class="form-group">
      <button class="btn btn-success btn-lg btn-block" (click)="addEmploy
    </div>
  </div>
</div>
```

Go to `employee-create.component.ts` file and add the following code.

```
import { Component, OnInit, Input } from '@angular/core';
import { Router } from '@angular/router';
import { RestApiService } from '../shared/rest-api.service';
```

positronX.io

```
  @Input() employeeDetails = {
  constructor(public restApi: R
  ngOnInit() {}
  addEmployee(dataEmployee: any
    this.restApi.createEmployee
      this.router.navigate(['/e
    });
  }
}
```

By adding given above code in employee create component, we can easily create an employee by making an HTTP POST request via Angular component.

Send HTTP GET and DELETE Requests

In this section i am going to manage employees list which we have created above. I will be using our RESTful API service by sending `get()` and `delete()` request via our custom apis.

Add code in **employee-list.component.html** file.

```
<div class="container custom-container-2">
  <!-- 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>
```

positronX.io

```
          <tbody>
            <tr *ngFor="let emplo
              <td>{{ employee.id
              <td>{{ employee.nam
              <td>{{ employee.ema
              <td>{{ employee.pho
              <td>
                <span class="edit
                  >Edit</span>
                <span class="delete" (click)="deleteEmployee(employee.id)"
                  >Delete</span
                >
              </td>
            </tr>
          </tbody>
        </table>
      </div>
    </div>
  </div>
```

Add code in employee-list.component.ts file.

```
import { Component, OnInit } from '@angular/core';
import { RestApiService } from '../shared/rest-api.service';
@Component({
  selector: 'app-employee-list',
  templateUrl: './employee-list.component.html',
  styleUrls: ['./employee-list.component.scss'],
})
export class EmployeeListComponent implements OnInit {
  Employee: any = [];
  constructor(public restApi: RestApiService) {}
  ngOnInit() {
    this.loadEmployees();
  }
  // Get employees list
  loadEmployees() {
    return this.restApi.getEmployees().subscribe((data: {}) => {
      this.Employee = data;
```

positronX.io

```
          this.loadEmployees();
        });
      }
    }
  }
```

Update Data

I am going to send HTTP PUT Request in Angular to update current employee data in our little demo app, It's pretty simple just follow the following steps.

Update code in **employee-edit.component.html**:

```
<div class="container custom-container">
  <div class="col-md-12">

    <h3 class="mb-3 text-center">Update Employee</h3>
    <div class="form-group">
      <input type="text" [(ngModel)]="employeeData.name" class="form-cont
    </div>
    <div class="form-group">
      <input type="text" [(ngModel)]="employeeData.email" class="form-con
    </div>
    <div class="form-group">
      <input type="text" [(ngModel)]="employeeData.phone" class="form-con
    </div>
    <div class="form-group">
      <button class="btn btn-success btn-lg btn-block" (click)="updateEmp
    </div>

  </div>
</div>
```

employee-edit.component.ts

```
    templateUrl: './employee-edit
    styleUrls: ['./employee-edit.
  })
  export class EmployeeEditCompon
    id = this.actRoute.snapshot.p
    employeeData: any = {};
    constructor(
      public restApi: RestApiServ
      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(d
          this.router.navigate(['/employees-list'])
        })
      }
    }
  }
```

Now you can test your Angular HttpClient application in the browser, just type `ng serve` in the terminal.

Run Angular Application

Start your project using given below command.

```
ng serve --open
```

positronX.io

Download the full code of this tuto

Have a good day, Keep learning.

Digamber

I am Digamber, a full-stack developer and fitness aficionado. I created this site to bestow my coding experience with newbie programmers. I love to write on JavaScript, ECMAScript, React, Angular, Vue, Laravel.

Twitter GitHub

Recommended Posts:

Angular 15 Bind Select Element to Object Tutorial

Angular 15 Capture Images from System Webcam Tutorial

How to Create Server Side Pagination in Angular 15 App

How to Show Hide Div on Radio Button Click in Angular 15

Angular 15 Detect Width and Height of Screen Tutorial

positronX.io

Angular 10 Digit Mobile Nu

Angular Detect Browser Na

Angular 15 Display JSON Data in Table Tutorial

Angular 15 FullCalendar Create and Display Dynamic Events

Angular 15 Image Upload, Preview, Crop, Zoom Example

positronX.io

Explore

About

Advertisement

Affiliate Disclosure

Privacy Policy

Disclaimer

Contact

Categories

Angular

Vue

React

Ionic

Laravel

Codeigniter

PHP

HTML 5, CSS 3, JavaScript, PHP, React, Angular, Laravel — Aimed to offer custom coding solutions on almost every modern programming language.