



## Bài 6: Gọi API trong Angular với HttpClient

### 1. Giới thiệu về HttpClient trong Angular

Trong Angular, để gọi API, chúng ta sử dụng **HttpClient**, một dịch vụ được tích hợp sẵn giúp chúng ta thực hiện các HTTP request như **GET, POST, PUT, DELETE** một cách dễ dàng.

Các lợi ích khi sử dụng **HttpClient**:

- Cung cấp phương thức dễ dàng để gọi API (RESTful API).
- Hỗ trợ **Observable** từ RxJS giúp xử lý bất đồng bộ.
- Hỗ trợ **interceptors** để thao tác với request hoặc response.
- Cung cấp tính năng **error handling** giúp xử lý lỗi API một cách linh hoạt.

### 2. Cài đặt HttpClientModule

Trước khi sử dụng **HttpClient**, bạn cần import **HttpClientModule** vào module chính (**app.module.ts**).

#### Bước 1: Import HttpClientModule

Mở file **src/app/app.module.ts** và thêm **HttpClientModule**:

typescript

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppComponent } from './app.component';
import { HttpClientModule } from '@angular/common/http'; // Import
HttpClientModule

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    HttpClientModule // Thêm HttpClientModule vào imports
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

---

### 3. Tạo Service để gọi API

---

Trong Angular, chúng ta nên tạo một **service** để quản lý logic gọi API, giúp mã nguồn dễ bảo trì và tái sử dụng.

#### Bước 2: Tạo Service

---

Chạy lệnh sau trong terminal để tạo service:

```
sh
```

```
ng generate service services/post
```

Lệnh này sẽ tạo ra **post.service.ts** trong thư mục **src/app/services/**.

---

## 4. Viết hàm gọi API

---

Mở file `src/app/services/post.service.ts` và thêm code để gọi API.

### Bước 3: Sử dụng HttpClient để gọi API

---

Chúng ta sẽ gọi API từ **JSONPlaceholder** để lấy danh sách bài viết (`posts`).

typescript

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class PostService {
  private apiUrl = 'https://jsonplaceholder.typicode.com/posts'; // URL API
  già lập

  constructor(private http: HttpClient) {}

  // Lấy danh sách bài viết
  getPosts(): Observable<any[]> {
    return this.http.get<any[]>(this.apiUrl);
  }
}
```

---

## 5. Sử dụng Service trong Component

---

Bây giờ, chúng ta sẽ sử dụng service này trong `AppComponent` để hiển thị danh sách bài viết.

### Bước 4: Inject Service vào Component

---

Mở `src/app/app.component.ts` và cập nhật như sau:

typescript

```
import { Component, OnInit } from '@angular/core';
import { PostService } from '../services/post.service';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent implements OnInit {
  posts: any[] = [];

  constructor(private postService: PostService) {}

  ngOnInit(): void {
    this.postService.getPosts().subscribe(
      (data) => {
        this.posts = data;
      },
      (error) => {
        console.error('Lỗi khi lấy dữ liệu:', error);
      }
    );
  }
}
```

---

## 6. Hiển thị dữ liệu lên giao diện

---

Mở file `src/app/app.component.html` và thêm đoạn mã sau để hiển thị danh sách bài viết:

html

```
<div class="container">
  <h2>Danh sách bài viết</h2>
  <ul *ngIf="posts.length > 0; else loading">
    <li *ngFor="let post of posts">
      <h4>{{ post.title }}</h4>
      <p>{{ post.body }}</p>
    </li>
  </ul>
  <ng-template #loading>
    <p>Đang tải dữ liệu...</p>
  </ng-template>
</div>
```

---

## 7. Kiểm tra kết quả

---

Chạy ứng dụng bằng lệnh:

```
sh
```

```
ng serve
```

Mở trình duyệt tại <http://localhost:4200>, bạn sẽ thấy danh sách bài viết được lấy từ API.

---

## 8. Thêm chức năng tạo bài viết mới (POST request)

---

Bây giờ, chúng ta sẽ thêm một form để tạo bài viết mới và gửi dữ liệu lên server.

---

### Bước 5: Thêm phương thức POST vào Service

---

Cập nhật `post.service.ts`:

typescript

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class PostService {
  private apiUrl = 'https://jsonplaceholder.typicode.com/posts';

  constructor(private http: HttpClient) {}

  getPosts(): Observable<any[]> {
    return this.http.get<any[]>(this.apiUrl);
  }

  createPost(postData: any): Observable<any> {
    return this.http.post<any>(this.apiUrl, postData);
  }
}
```

## Bước 6: Thêm Form nhập bài viết mới

---

Mở `app.component.ts` và thêm hàm tạo bài viết:

typescript

```
import { Component, OnInit } from '@angular/core';
import { PostService } from '../services/post.service';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent implements OnInit {
  posts: any[] = [];
  newPost = { title: '', body: '' };

  constructor(private postService: PostService) {}

  ngOnInit(): void {
    this.postService.getPosts().subscribe((data) => {
      this.posts = data;
    });
  }

  addPost(): void {
    this.postService.createPost(this.newPost).subscribe((post) => {
      this.posts.unshift(post); // Thêm bài viết vào danh sách
      this.newPost = { title: '', body: '' }; // Reset form
    });
  }
}
```

## Bước 7: Thêm Form nhập dữ liệu vào giao diện

---

Cập nhật `app.component.html`:

html

```
<div class="container">
  <h2>Danh sách bài viết</h2>

  <form (ngSubmit)="addPost()">
    <input type="text" [(ngModel)]="newPost.title" name="title"
placeholder="Tiêu đề bài viết" required>
    <textarea [(ngModel)]="newPost.body" name="body" placeholder="Nội dung
bài viết" required></textarea>
    <button type="submit">Thêm bài viết</button>
  </form>

  <ul>
    <li *ngFor="let post of posts">
      <h4>{{ post.title }}</h4>
      <p>{{ post.body }}</p>
    </li>
  </ul>
</div>
```

Đừng quên import `FormsModule` vào `app.module.ts`:

typescript

```
import { FormsModule } from '@angular/forms';

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

---

## 9. Tổng kết

---



- Học cách sử dụng `HttpClientModule` để gọi API.
  - Tạo Service (`post.service.ts`) để quản lý việc gọi API.
  - Thực hiện **GET** để lấy danh sách bài viết.
  - Thực hiện **POST** để thêm bài viết mới.
  - Hiển thị dữ liệu lên giao diện bằng `*ngFor`.
  - Dùng `ngModel` để tạo form nhập bài viết mới.
- 

👉 Bài tiếp theo: **Interceptor trong Angular để quản lý request và response.**