

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Angular giao tiếp với Backend
- ✓ Sử dụng module http service để làm việc với Backend
- ✓ Tạo và sử dụng service cho yêu cầu với http

PHẦN I

Bài 1 (2 điểm)

Sử dụng Api của học phần NodeJs và Http request Angular thực hiện hiển thị danh sách bài viết – post trên component post

The screenshot shows a web application titled "FPoly Product Manager" with a navigation bar containing "Home", "Product list", "Manager Blog list", and a "Manager" dropdown. The main content area is titled "Quản lý bài viết" (Manage articles). It features a table with columns: "#", "Tiêu đề" (Title), "Ngày cập nhật" (Update date), and "Thao tác" (Action). The table contains two rows of data. To the right of the table is a form titled "Thêm - sửa bài viết" (Add - edit article) with a text input for "Tiêu đề bài viết" (Article title) and a text area for "Nội dung bài viết" (Article content). A green "Lưu" (Save) button is at the bottom of the form.

#	Tiêu đề	Ngày cập nhật	Thao tác
23	Front end VueSJ	Front end VueSJ	Xoá
24	Front end Angular 10	Có thể làm được mọi thứ	Xoá

Thêm - sửa bài viết

Tiêu đề bài viết

Nội dung bài viết

Lưu

Hướng dẫn

Tạo component post và thiết kế giao diện như mẫu

Cài đặt Http module tại app.module.ts:

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

```
imports: [
```

```
  BrowserModule,
```

```
  AppRoutingModule,
```

```
  FormsModule,
```

```
  ReactiveFormsModule,
```

```
  HttpClientModule
```

]

Đọc danh sách bài viết với phương thức GET của đối tượng HttpClient

```
ngOnInit(): void {  
  this.getAllPost();  
  
}  
  
getAllPost(){  
  this.httpService.get(this.url).subscribe(data=>{  
    this.posts=data.posts;  
  })  
}
```

Hiển thị bài viết ra ngoài giao diện

```
<table class="table">  
  
  <tr>  
    <th scope="col">#</th>  
    <th scope="col">Tiêu đề</th>  
    <th scope="col">Ngày cập nhật</th>  
    <th scope="col">Thao tác</th>  
  
  </tr>  
  
  <tr *ngFor="let post of posts">  
    <td>{{post.postId}}</td>  
    <td>{{post.title}}</td>  
    <td>{{post.content}}</td>  
    <td>Xoá</td>  
  
  </tr>  
  
</table>
```

Xem kết quả:localhost:4200/posts

Bài 2 (2 điểm)

Thực hiện thêm 1 bài viết mới

FPPoly Product Manager
Home
Product list
Manager Blog list
Manager

Quản lý bài viết

#	Tiêu đề	Ngày cập nhật	Thao tác
23	Front end VueSJ	Front end VueSJ	Xoá
24	Front end Angular 10	Có thể làm được mọi thứ	Xoá

Thêm - sửa bài viết

Tiêu đề bài viết

Nội dung bài viết

Lưu

Hướng dẫn

Thực hiện binding dữ liệu 2 chiều trên form

```
<div class="col-3">
  <h3>Thêm - sửa bài viết</h3>
  <form>
    <div class="form-group">
      <input type="text" placeholder="Tiêu đề bài viết" [(ngModel)]="post.title" name="title">
    </div>
    <div class="form-group">
      <textarea [(ngModel)]="post.content" name="content" cols="30" rows="10" placeholder="Nội dung bài viết"></textarea>
    </div>
    <div class="form-group btnX">
      <button class="btn btn-success center" (click)="addPost()">Lưu</button>
    </div>
  </form>
</div>
```

Tại post.component.ts thực hiện code

```
post: IPost;
onCreatePost(dataPost: IPost){
  this.httpService.post(this.url, dataPost).subscribe(p=>{
    console.log(p);
  })
}
addPost(){
  this.onCreatePost(this.post);
  this.getAllPost();
}
```

Kết quả: như hình trên

Bài 3 (2 điểm)

Thực hiện thao tác thiết kế nút xóa và code cho nút này

The screenshot shows a web application titled "FPoly Product Manager" with a navigation bar containing "Home", "Product list", "Manager Blog list", and a "Manager" dropdown. The main content area is titled "Quản lý bài viết" (Manage posts). It features a table with three columns: "#", "Tiêu đề" (Title), "Ngày cập nhật" (Update date), and "Thao tác" (Action). The table contains three rows of data. To the right of the table is a form titled "Thêm - sửa bài viết" (Add - edit post) with a text input for "Tiêu đề bài viết" (Post title) and a text area for "Nội dung bài viết" (Post content). A green "Lưu" (Save) button is at the bottom right of the form.

#	Tiêu đề	Ngày cập nhật	Thao tác
1	Front end VueSJ	Front end VueSJ	Xoá
2	Front end Angular 10	Có thể làm được mọi thứ	Xoá
3	PHP1	Làm quen với lập trình web	Xoá

Hướng dẫn

Bổ sung phần Html như sau

```
<tbody *ngIf="posts.length>=1">
  <tr *ngFor="let post of posts; let i=index">
    <td>{{i+1}}</td>
    <td>{{post.title}}</td>
    <td>{{post.content}}</td>
    <td>
      <button class="btn btn-danger" (click)="onDelete(post.postId)"> Xoá </button>
    </td>
  </tr>
</tbody>
```

Hàm onDelete viết tại Component.ts:

```
onDelete(id:number){
  this.httpService.delete(this.url+id).subscribe(p=>{
    console.log(p);
    this.getAllPost();
  })
}
```

Chạy và kiểm tra kết quả

PHẦN II

Bài 4 (2 điểm)

Thực hiện các chức năng thêm, xoá, sửa post với service cho http request

Hướng dẫn:

- Tạo service
ng generate service post

- Code cho service post

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { IPost } from './entities/post';

@Injectable({
  providedIn: 'root'
})
export class PostService {

  url='http://127.0.0.1:3000/blog/posts/';

  constructor(private httpService:HttpClient) { }

  storePost(postData:IPost){
    this.httpService.post(this.url,postData).subscribe(p=>{
      console.log(p);
    })
  }

  getAllPost(){
    return this.httpService.get(this.url);
  }

  deletePost(id:number){
    return this.httpService.delete(this.url+id);
  }
}
```

- Sử dụng service tại post.component.ts

```
ngOnInit(): void {
  this.getAllPost();
}

getAllPost(){
  this.postService.getAllPost().subscribe(data=>{
    this.posts=data.posts;
    console.log(this.posts);
  })
}

addPost(){
  this.postService.storePost(this.post);
}
```

```
onDelete(id:number){
  this.postService.deletePost(id).subscribe(p=>{
    console.log(p);
    this.getAllPost();
  })
}
```

- Chạy và kiểm tra kết quả xử lý khi nhấn submit với console của trình duyệt

Bài 5 (2 điểm)

Thực hiện Quản lý lỗi xảy ra khi thực hiện các thao tác thêm, xóa, sửa

Xây dựng trạng thái loading... khi ứng dụng đang tải dữ liệu

```
<tbody *ngIf="posts.length>=1 && !isFetching">
  <tr *ngFor="let post of posts; let i=index">
    <td>{{i+1}}</td>
    <td>{{post.title}}</td>
    <td>{{post.content}}</td>
    <td><button class="btn btn-danger" (click)="onDelete('1234')">Xóa</button></td>
  </tr>
</tbody>
<p *ngIf="isFetching">Loading....</p>
```

```
getAllPost(){
  this.isFetching=true;
  this.postService.getAllPost().subscribe(data=>{
    this.isFetching=false;
    this.posts=data.posts;
    console.log(this.posts);
  },error=>{
    this.error=error.message;
  })
}
```

*** Yêu cầu nộp bài:

SV nén file (hoặc share thư mục google drive) bao gồm các yêu cầu đã thực hiện trên, nộp LMS đúng thời gian quy định của giảng viên. KHÔNG NỘP BÀI COI NHƯ KHÔNG CÓ ĐIỂM.

--- Hết ---