

# Lab 06 HTTP

## Intro

ในแลปนี้เราจะสร้าง Web App ที่ใช้งาน HTTP GET, POST และ DELETE ซึ่งเป็น Method ที่ใช้บ่อยที่สุด ผู้ที่ใช้ Web App นี้จะสามารถเลือกการกระทำได้สามแบบได้แก่ ดูข้อความ โปสข้อความ และ ลบข้อความ (ลักษณะคล้ายกับ Twitter แบบง่าย) ก่อนอื่นเลยเราจะต้องกำหนดขอบเขตของข้อมูลที่เราจะเก็บ โดยในครั้งนี้เราจะเก็บข้อมูลชื่อผู้โพส ตัวข้อความ และเวลาที่โพส เพื่อให้ข้อมูลทุกส่วนเป็นไปในทางเดียวกัน เราจะสร้างคลาสที่เก็บข้อมูลดังนี้

```
export class Tweet {  
  id : number;  
  name : string;  
  msg : string;  
  date : Date;  
}
```

## Setup Mock Server

เพื่อให้ง่ายต่อการ Debug เราจะสร้าง Mock Server ในรูปแบบของ Service โดยการ Implements InMemoryDbService ในฟังก์ชัน createDb เราอาจจะสร้างข้อมูลจำลอง 1 entry เพื่อเอาไว้เป็น Default ตอนโหลดแอปครั้งแรก

1. ดูการสร้าง InMemoryDbService ใน Lab05
2. ถ้าหาก StackBlitz แจ้งว่าไม่ได้ลง angular-in-memory-web-api ให้คลิกที่ Install Package
3. อย่าลืม import Module ต่าง ๆ ในไฟล์ app.module.ts

## Tweet Service

Tweet Service จะเป็นส่วนที่เราใช้เชื่อมต่อกับฐานข้อมูล (ในที่นี้คือ Mock Server) ตอนนี้เราจะสร้างฟังก์ชัน getTweets ก่อนเพื่อให้ Display Tweet Component เรียกใช้งานได้ ซึ่งฟังก์ชันนี้จะใช้ HTTP GET โดยส่งผ่านแค่ URL

1. เลียนแบบการสร้าง Tweet Service ได้จาก Product Service ของ Lab05
2. URL ที่ใช้จะเป็นในลักษณะของ "api/ชื่อตัวแปรในฟังก์ชัน createDb"

## Display Tweet Component

**Display Tweet Component** เป็นส่วนที่แสดงผล **Tweet** ทั้งหมด หลัก ๆ คือการเรียกใช้งาน **Tweet Service** ใน **ngOnInit** เพื่อเอาข้อมูลมาเก็บไว้ในตัวแปรภายใน **Component** เราสามารถเลียนแบบการสร้าง **Component** นี้ได้จาก **Product List Component** ใน **Lab05**

## Add Tweet Component

**Add Tweet Component** เป็นส่วนที่ใช้โพสข้อความหรือเพิ่ม **Tweet** ที่เก็บอยู่ใน **Mock Server** ก่อนอื่นเราจะต้องสร้างฟังก์ชัน **addTweet** ใน **Tweet Service** ก่อน ผ่านทางการใช้งาน **HTTP POST** เนื่องจากเราจะต้องส่งข้อมูลเพิ่มเติมทำให้รูปแบบการใช้งานจะต่างจาก **HTTP GET** เล็กน้อย **HTTP POST** จะรับ 3 **Parameters** ได้แก่ **URL, Data, และ Option**

1. **URL** ใช้แบบเดียวกับ **HTTP GET** เนื่องจากเราจะเพิ่มข้อมูลไปยังตัวแปรเดิม
2. **Data** เป็นข้อมูลที่จะเพิ่ม ในที่นี้คือ **Tweet Object** ซึ่งจะมีข้อมูลชื่อผู้ใช้และข้อความที่รับเข้ามาจากแบบฟอร์ม (ที่เราจะสร้างในลำดับถัดไป) ส่วน **id** จะต้อง **Generate** เอง ดังนั้นเราควรสร้าง **private variable** เพื่อจัดการไม่ใช่ **id** ซ้ำกัน ส่วนเวลาที่โพสเราสามารถใส่ฟังก์ชัน **new Date()** จาก **JavaScript** ได้โดยตรง

```
let tweet : Tweet = {  
  id:this.id,  
  name:p1,  
  msg:p2,  
  date:new Date()  
};
```

3. **Option** จะเป็นการระบุรูปแบบการส่งข้อมูลรวมทั้ง **Header** ด้วย

```
const httpOptions = {  
  headers: new HttpHeaders({  
    'Content-Type': 'application/json',  
    'Authorization': 'my-auth-token'  
  })  
};
```

4. การเรียกใช้ HTTP GET ก็จะเป็นในลักษณะคล้ายแบบนี้

```
this.http.post<Tweet>(this.tweetUrl, tweet, httpOptions);
```

หลังจากที่เราเพิ่มฟังก์ชันใน Tweet Service เราก็พร้อมที่จะสร้าง Add Tweet Component ซึ่ง Component จะมีส่วนที่รับ Input มาจากผู้ใช้นั้นก็คือ Form นั่นเอง เราจะใช้ Reactive Form Module ในการควบคุม Form ทั้งหมด

1. ที่ไฟล์ app.module.ts ให้ import ReactiveFormsModule จาก @angular/forms รวมทั้งเพิ่ม ReactiveFormsModule ในส่วน imports ของ NgModule

2. สร้าง Add Tweet Component

a. Import FormControl, FormGroup, Validators จาก @angular/forms

b. Import TweetService และ Tweet จากไฟล์ที่เกี่ยวข้อง

c. Inject private tServ : TweetService ใน constructor

d. สร้างตัวแปร Form Group แล้วเพิ่ม Form Control 2 ตัว

เอาไว้ควบคุมข้อมูลชื่อผู้ใช้และข้อความที่พิมพ์

```
form = new FormGroup({  
  name : new FormControl(''),  
  msg : new FormControl(''),  
});
```

e. สร้างฟังก์ชัน onSubmit เอาไว้ผูกกับปุ่ม Tweet เพื่อโพสข้อความ ซึ่งด้านในก็เรียกใช้ฟังก์ชัน addTweet ของ Tweet Service

```
onSubmit() {  
  this.tServ.addTweet(  
    this.form.value.name,  
    this.form.value.msg  
  ).subscribe();  
}
```

f. สร้าง Form เพื่อรับข้อมูลจากผู้ใช้ ตั้งชื่อฟอร์มเป็นชื่อตัวแปรของ FormGroup ส่วน Input อื่น ๆ ให้ตั้งค่า FormControlName เป็นชื่อตัวแปรของ FormControl ที่สร้างไว้ใน

FormGroup นอกจากนี้แล้วเราจะต้องกำหนดค่า (ngSubmit) ให้เป็นชื่อฟังก์ชัน onSubmit ที่เราสร้างขึ้นเมื่อสักครู่นี้

```
<form [formGroup]="form" (ngSubmit)="onSubmit()">
  Name :
  <input type="text" formControlName="name"><br>
  Message :
  <textarea formControlName="msg"></textarea>
  <div><input type="submit" value="Tweet"></div>
</form>
```

หลักจากที่เราสร้าง Add Tweet Component เสร็จแล้ว เราก็จะเพิ่ม Component นี้ให้กับแอปหลักโดยการสร้างเมนูและใช้ routerLink เป็นตัวเชื่อมแต่ละ Component

1. ใน app.module.ts ให้ import RouterModule จาก @angular/router
2. ภายใน NgModule ใต้ imports เพิ่ม RouterModule พร้อมกำหนด path

```
RouterModule.forRoot([
  { path: '', component: DisplayTweetComponent },
  { path: 'tweet', component: AddTweetComponent }
]),
```

3. สร้างเมนูใน app.component.html พร้อมกำหนด routerLink

```
<h1>Twitter</h1>
| <a routerLink="">Timeline</a>
| <a routerLink="/tweet">Tweet</a> |
<router-outlet></router-outlet>
```

## Delete Tweet Feature (แบบฝึกหัด)

ในส่วน **Display Tweet Component** ให้เพิ่มปุ่ม **Delete** สำหรับแต่ละ **Tweet** เมื่อคลิกแล้วให้เรียกใช้ฟังก์ชันที่ลบ **Tweet** นั้น โดยใช้ **id** เป็นตัวระบุ **Tweet** การใช้งาน **HTTP DELETE** จะมีอยู่ **2 Parameters** คือ **URL** และ **Option** ในส่วนของ **URL** จะอยู่ในรูปแบบของ “**api/ชื่อตัวแปรใน createDb/ค่า id ของ Tweet ที่ต้องการลบ**” หลังจากลบแล้วอย่าลืมอัปเดต **Tweet** ที่แสดงผลด้วย

```
return this.http.delete(url,httpOptions);
```