

# Upvote

## Re-factor Display Tweet Component

เราต้องการแยกการทำงานของ DisplayTweet ออกเป็น 2 ส่วนคือ (1) DisplayTweet ที่แสดงผลแค่ Tweet เดียว และ (2) Timeline ที่จัดการ DisplayTweet ทั้งหมด อีกครั้งหนึ่ง

1. สร้าง Timeline Component
2. คัดลอก Code บางส่วนมาจาก DisplayTweet
  - a. นำเข้า Tweet และ Tweet Service
  - b. Inject Tweet Service
  - c. ประกาศตัวแปร tweets : Tweet[]
  - d. Subscribe ฟังก์ชัน getTweets ของ Tweet Service

```
import { Component, OnInit } from '@angular/core';
import { Tweet } from '../tweet';
import { TweetService } from '../tweet.service';

@Component({
  selector: 'app-timeline',
  templateUrl: './timeline.component.html',
  styleUrls: ['./timeline.component.css']
})
export class TimelineComponent implements OnInit {

  constructor(
    private tServ : TweetService
  ) { }

  tweets : Tweet[];

  ngOnInit() {
    this.tServ.getTweets().subscribe(val => this.tweets = val);
  }
}
```

3. แก้ไข DisplayTweet Component
  - a. ลบ <div> ที่มี \*ngFor
  - b. ลบ tweets และฟังก์ชันที่เกี่ยวข้อง
  - c. ลบ auth และคำสั่งที่เกี่ยวข้อง

- d. นำเข้า Input เพิ่มจาก Component, OnInit
- e. ประกาศตัวแปร tweet : Tweet แบบ @Input()
- f. นำเข้า Time Ago Pipe และประกาศ TimeAgoPipe ใน declarations ของ App Module

```

<div class="tweet">
  <b>{{tweet.name}}</b> · {{tweet.date | timeAgo}}
  <p>{{tweet.msg}}</p>
</div>

import { Component, OnInit, Input } from '@angular/core';
import { Tweet } from '../tweet';

@Component({
  selector: 'app-display-tweet',
  templateUrl: './display-tweet.component.html',
  styleUrls: ['./display-tweet.component.css']
})
export class DisplayTweetComponent implements OnInit {

  constructor( ) { }

  @Input() tweet : Tweet;

  ngOnInit() {

  }

}

import { TimeAgoPipe } from 'time-ago-pipe';
declarations: [ ..., TimeAgoPipe ]

```

#### 4. เชื่อมโยง Timeline Component กับ DisplayTweet Component

```

<div *ngFor="let tweet of tweets">
  <app-display-tweet [tweet]="tweet"></app-display-tweet>
</div>

```

#### 5. แก้ไข path ใน App Module ให้ Timeline Component เป็น Root Component

```

RouterModule.forRoot([
  {path: '', component: TimelineComponent},
  {path: 'tweet', component: AddTweetComponent},
  {path: 'login', component: LoginComponent}
])

```

## Upvote Service and Component

เราต้องการให้ผู้ใช้สามารถ Upvote Tweet ได้ (เหมือน Like)

1. สร้าง Upvote Class โดยมีสมาชิกเป็น id:number และ [username:string]:number

```
export class Upvote {
  id: number;
  [username:string] :number;
}
```

2. แก้ไข InMemoryData Service โดยเพิ่มตัวแปร upvotes ในฐานข้อมูล

```
import { Injectable } from '@angular/core';
import { InMemoryDbService } from 'angular-in-memory-web-api';

import { Tweet } from './tweet';

@Injectable({
  providedIn: 'root',
})
export class InMemoryDataService implements InMemoryDbService {

  createDb() {
    const tweets : Tweet[] = [{
      id:0,
      name:'Guest',
      msg:'Hello',
      date:new Date(2020,1,14)
    }];
    const upvotes : Upvote[] = [{
      id:0,
      "Guest":0,
    }]
    return {tweets, upvotes};
  }
  constructor() { }
}
```

3. สร้าง Upvote Service (คล้าย Tweet Service)

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

import { Upvote } from './upvote';

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

```

};

@Injectable()
export class UpvoteService {

  constructor(
    private http : HttpClient
  ) { }

  private upvoteUrl = 'api/upvotes';

  getVote(id) : Observable<Upvote> {
    return this.http.get<Upvote>(this.upvoteUrl+'/'+id);
  }

  updateVote(upvote) {
    return this.http.post<Upvote>(this.upvoteUrl,upvote,httpOptions);
  }
}

```

#### 4. สร้าง Upvote Component

```

import { Component, OnInit, Input } from '@angular/core';
import { UpvoteService } from '../upvote.service';
import { AuthService } from '../auth.service';

@Component({
  selector: 'app-upvote',
  templateUrl: './upvote.component.html',
  styleUrls: ['./upvote.component.css']
})
export class UpvoteComponent implements OnInit {

  constructor(
    private uServ : UpvoteService,
    private auth : AuthService
  ) { }

  @Input() voteId : number;

  name;
  upvote;
  votes;
  userVote;
  isLoggedIn;

  ngOnInit() {

```

```

this.auth.isLoggedIn.subscribe(val => this.isLoggedIn = val);
this.auth.username.subscribe(val => this.name = val);
if (localStorage.getItem("isLoggedIn")==="true") {
  this.isLoggedIn = true;
  this.name = localStorage.getItem("username");
}
this.uServ.getVote(this.voteId).subscribe(val => {
  this.upvote = val
  if(this.name!==undefined) this.userVote = val[this.name]
  this.sum(val)
},
error => {
  this.votes = 0
  this.upvote = {id:this.voteId}
});

console.log(this.name)
}

sum(obj) {
  var sum = 0;
  for( var el in obj )
    if( obj.hasOwnProperty( el ) )
      sum += parseFloat( obj[el] );
  this.votes = sum - obj["id"];
}

vote() {
  if(this.userVote===undefined) {
    this.userVote = 1;
    this.upvote[this.name] = 1;
  } else {
    let newVote = this.userVote == 1 ? 0 : 1;
    this.userVote = newVote;
    this.upvote[this.name] = newVote;
  }

  this.uServ.updateVote(this.upvote).subscribe();
  this.sum(this.upvote);
}
}

```

```

{{votes}} <i class="arrow up" (click)="vote()" *ngIf="isLoggedIn"></i>

```

```

i {
  border: solid black;
  border-width: 0 3px 3px 0;
}

```

```

display: inline-block;
padding: 3px;
}

.up {
transform: rotate(-135deg);
-webkit-transform: rotate(-135deg);
}

```

## 5. เพิ่ม Upvote Component ไปยัง DisplayTweet Component

```

<div class="tweet">
  <b>{{tweet.name}}</b> · {{tweet.date | timeAgo}}
  <p>{{tweet.msg}}</p>
  <app-upvote [voteId]="tweet.id"></app-upvote>
</div>

```

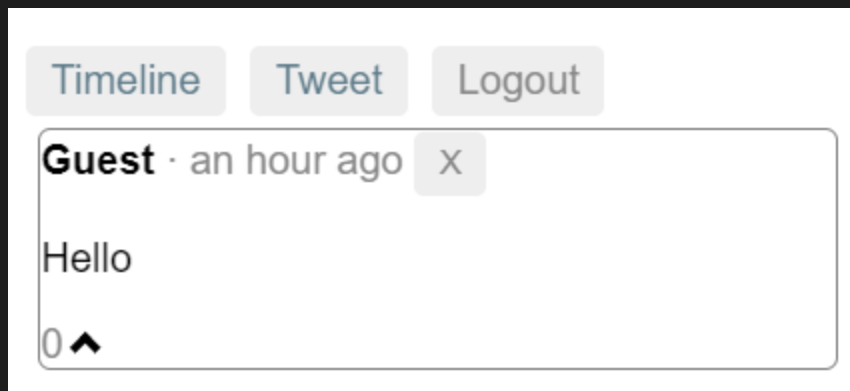
## Practice

1. แก้ไขปุ่ม Delete โดยการสร้าง Delete Component แล้วเพิ่มไปที่ Display Tweet Component ส่วนพีเจอร်ของปุ่ม Delete จะยังคงเดิม

```

<div class="tweet">
  <b>{{tweet.name}}</b> · {{tweet.date | timeAgo}}
  <app-delete [voteId]="tweet.id"></app-delete>
  <p>{{tweet.msg}}</p>
  <app-upvote [voteId]="tweet.id"></app-upvote>
</div>

```



2. แก้ไขสีของปุ่ม Upvote โดยถ้าผู้ใช้มีโหวตอยู่แล้วให้เป็นสีเขียว แต่ถ้าไม่มีโหวตให้เป็นสีดำ

1 → 2