Angular Cheat Sheets

การติดตั้ง

รายละเอียดวิธีการติดตั้ง
https://angular.io/guide/setup-local
ดาวน์โหลดและติดตั้งโปรแกรม Node.js (เวอร์ชัน LTS) จะมีโปรแกรม npm (เครื่องมือจัดการ package ของ node.js มาด้วย)
https://nodejs.org/en/
ติดตั้ง Angular framework
npm install -g @angular/cli
ตรวจสอบเวอร์ชันของ angular ที่ติดตั้งไป
ng --version

Angular CLI ที่สำคัญ

สร้างโปรเจ็คใหม่
ng new <ชื่อโปรเจ็ค>
Start การทำงานของ Angular Development Server
ng serve
คอมไพล์ Angular Application สำหรับ Deploy ขึ้น Server
ng build --prod
สร้าง Component ใหม่
ng g component <ชื่อComponent> --skipTests
ชื่อ Component จะถูกสร้างเป็น folder ดังนั้นใส่ / คั่นเพื่อสร้างเป็น folder ย่อยได้
ng g component <ชื่อโฟร์เดอร์/ชื่อComponent> --skipTests

ติดตั้ง BootStrap และ PrimeNG Theme ใน Angular

ดาวน์โหลดและติดตั้ง BootStrap ในโปรเจ็ค
npm install bootstrap --save
เพิ่มใน angular.json ที่ build->style
"node_modules/bootstrap/dist/css/bootstrap.min.css",

ดาวน์โหลดและติดตั้ง PrimeNG ในโปรเจ็ค
npm install primeng --save *npm install primeicons --save* *npm install @angular/animations --save*
เพิ่มใน angular.json ที่ build->style
"node_modules/primeicons/primeicons.css", *"node_modules/primeng/resources/themes/nova-light/theme.css",* *"node_modules/primeng/resources/primeng.min.css",*

โครงสร้างโฟร์เดอร์และไฟล์ที่สำคัญ

Folder *src/app* เก็บ Source code ของโปรเจ็ค
ใน Folder app
app.module.ts คือไฟล์กำหนดค่าการอ้างถึงโมดูลต่างๆ .css คือ style sheet .html คือ UI template .ts คือ Source code

การสร้าง Routes and Navigation

ใช้คลาส Routes สร้าง Object ในรูปแบบของอารเรย์

```
16  import { Routes, RouterModule } from '@angular/router';
17  const appRoutes: Routes = [
18    {path: '', component: AppComponent},
19    {path: 'login', component: LoginComponent},
20    {path: 'member', component: MemberComponent},
21  ]
```

Register Routes ที่สร้างขึ้นเข้าไปที่ Application

```
34    imports: [
35      RouterModule.forRoot(appRoutes),
```

กำหนดพื้นที่แสดงผลใน Application

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

สร้างลิงค์โดยใช้ RouterLink ใน tag a (ขึ้นต้นด้วย / เสมอ)

```
<a routerLink="/login">Login</a>
```

การส่งพารามีเตอร์ไปกับ Route ใช้เครื่องหมาย :

```
{path: 'member/:id', component: MemberComponent},
```

การรับพารามีเตอร์ที่ส่งมากับ Route

```
import { ActivatedRoute } from '@angular/router';
```

```
constructor( private route : ActivatedRoute) {}
```

```
let id = this.route.snapshot.params['id'];
```

เปลี่ยนหน้าหรือโหลด Component โดยการเขียนโปรแกรมใน .ts

```
import { Router } from '@angular/router';
```

```
constructor(private router : Router) {
}
```

```
this.router.navigateByUrl('/member');
```

Data Passing and Data Binding

สร้าง Service ใหม่
ng g service <ชื่อ Service> --skipTests

การกำหนด properties/attributes ใน Service

```
1  import { Injectable } from '@angular/core';
2
3  @Injectable({
4    providedIn: 'root'
5  })
6  export class DatapassService {
7    public username;
8    public password;
9    constructor() { }
10 }
```

การเรียกใช้ Service ในไฟล์ .ts

```
import { DatapassService } from '../datapass.service';
```

```
constructor(private data : DatapassService,
  private route : ActivatedRoute) {}
```

```
console.log(this.data.username);
```

การทำ One way data binding
แสดงค่าตัวแปรใน .html *{{ชื่อตัวแปร}}*
ต้องมีตัวแปรที่เป็น attribute ใน .ts

การทำ Two ways data binding
เรียกใช้ FormModule *import {FormsModule} from '@angular/forms';*
เรียกใช้ Add FormModule เข้าไปในโปรเจ็ค *Imports FormsModule*
ผูก attribute ของ Tag input *[(ngModel)]="ชื่อตัวแปร"*
ต้องมีตัวแปรที่เป็น attribute ใน .ts

Directive *ngIf and *ngFor

การตรวจสอบเงื่อนไขใน HTML โดยใช้ *ngIf
*<div *ngIf="Condition">Content *
*<div *ngIf="Condition; else elseblock">* *Content true* * * *<ng-template #elseblock>* *Content else* *</ng-template>*

การวนลูปแสดงผลใน HTML โดยใช้ *ngFor
*<div *ngFor="let item of items; let i = index;">* *{{i}}. {{item}}* * *

Observable และ Async/Await ในการ request http

รูปแบบการเรียกใช้ Observable

POST

```
this.http.post('<URL>', JSON.stringify(<jsonstring>))
      .subscribe(data =>{
            console.log(data);
       }, error =>{
           console.log(error);
       });
```

GET

```
this.http.get('<URL>').subscribe(data =>{
            console.log(data);
       }, error =>{
           console.log(error);
       });
```

Async/Await (Await ต้องเรียกอยู่ใน Async เมธอดเท่านั้น)

การเรียก GET / POST แบบ Async/Await

```
async getUser(){
 let response = await this.http.get('<URL>').toPromise();
 return response;
}
```

การเรียกใช้

```
async ngOnInit() {
 let response : any = await this.getUser();
 console.log(response);
 this.fullname = response.firstName;
}
```

การแปลงไฟล์เป็น Base64 และ Upload

Input file ใน HTML

```
<input type="file" name="" id=""
(change)="getFile($event.target.files)"> <br>
<button (click)="upload()">Upload file</button>
```

เมธอดการแปลงไฟล์เป็น Base64

```
getFile(files : FileList){
    let base64;
    let file = files.item(0);
    let filename = file.name;
    let reader = new FileReader();
    reader.readAsDataURL(file);
    reader.onload = () => {
      base64 = reader.result;
    };
}
```