

Lab02 – Angular Module & Component

Brand New App

อาทิตย์ที่แล้วเราได้ลองทำ Web Application จากการ Fork โปรเจคที่มีอยู่แล้ว ในวันนี้เราจะลองสร้าง Web Application จากศูนย์ ก่อนอื่นให้เราเข้าเว็บ stackblitz และ signin ใต้ชื่อของเรา

1. กดที่ปุ่ม START A NEW APP จากนั้นเลือก Angular แก้ไขชื่อ Repository เป็น รหัสนักศึกษาlab02 (ตัวพิมพ์เล็กทั้งหมด)
2. เปิดเข้าไปดูไฟล์ app.module.ts เราจะสังเกตเห็นว่าไฟล์ Module ได้ถูกสร้างให้เราเรียบร้อยแล้ว

Module ถือเป็นหัวใจสำคัญของ Application เนื่องจากมันเป็นตัวรวม Components, Directives, Pipes และ Services ซึ่งเป็นส่วนย่อยของ Application เพราะฉะนั้นคำสั่งส่วนมากที่อยู่ใน Module จะเป็นการนำเข้าส่วนย่อยต่าง ๆ เข้ามา

เราสามารถมองการทำงานของ Module ในรูปแบบ Class ได้ ใน Class เรามีการประกาศฟังก์ชันแบบ Public และ Private ฟังก์ชันแบบ Public คือ API (Application Programming Interface) ที่อนุญาตให้ส่วนอื่น ๆ ของโค้ดเราใช้งานได้ ส่วน ฟังก์ชันแบบ Private ก็เน้นไปในด้านการทำงานภายในที่จะถูกซ่อนไว้ ในรูปแบบเดียวกัน Module สามารถ export หรือซ่อน Components, Directives, Pipes และ Services ได้ เพราะฉะนั้นส่วนไหนที่ export ก็จะสามารถนำไปใช้งานที่อื่นได้

ก่อนอื่นเลยเราต้องนำเข้า NgModule ถึงจะใช้งาน Module ได้

```
import { NgModule } from '@angular/core';
@NgModule({
  imports: [ ... ],
  declarations: [ ... ],
  bootstrap: [ ... ]
})
export class AppModule { }
```

ตัวอย่างข้างต้นเป็นรูปการแปลง Class ที่ชื่อ AppModule ให้กลายเป็น Module ด้วยการใช้ NgModule โดย NgModule จะมีการบังคับให้เรากำหนด Properties ทั้งสามคือ

1. Import ประกอบไปด้วย Module หลาย ๆ ตัว
2. Declaration ประกอบไปด้วย Components, Directives และ Pipes
3. Bootstrap ประกอบไปด้วย Component หลัก หรือ Root Component

ตอนนี้ Module หลักของแอปเรามีการนำเข้า 2 Module ย่อยคือ BrowserModule และ FormsModule เรียกใช้งาน 2 Component คือ AppComponent และ HelloComponent โดยมี AppComponent เป็น Root Component

List Movies

1. คลิกขวาที่โฟลเดอร์ App แล้วเลือก New File ตั้งชื่อเป็น movies.ts
ไฟล์นี้จะเป็นไฟล์ที่เก็บข้อมูลภาพยนตร์ของเรา
2. แก้ไขไฟล์นี้โดยก็สร้างตัวแปรชนิด Array ชื่อว่า movies ตามรูปแบบต่อไปนี้

```
export const movies = [  
  {  
    name: 'Titanic',  
    year: 1997,  
    genre: 'drama'  
  }  
];
```

3. สร้าง Component ใหม่โดยคลิกขวาที่โฟลเดอร์ App แล้วเลือก Angular Generator ตามด้วย Component ตั้งชื่อเป็น movie-list
4. ไปที่ไฟล์ movie-list.component.html แล้วแก้ไขเป็น

```
<h1>  
Movies  
</h1>
```

- สังเกตว่าข้อความใน Component นี้ไม่ได้แสดงผลใน Display นั่นเป็นเพราะเราไม่ได้เรียกใช้งาน Component ในแอปหลัก ให้กลับไปไฟล์ app.component.html แล้วเพิ่ม Tag ต่อไปนี้
- เราต้องการแสดงรายชื่อหนังโดยเริ่มจากการนำเข้าไฟล์ข้อมูลหนังมายัง Component ที่สร้างขึ้นใหม่ ไปที่ไฟล์ movie-list.component.ts แล้วเพิ่มบรรทัดนี้

```
import { movies } from '../movies';
```

- ทำการ Bind ตัวแปรที่นำเข้ามากับตัวแปรใหม่ที่จะใช้ตอนแสดงผล โดยการเพิ่มบรรทัดนี้

```
export class MovieListComponent implements OnInit {
  ms = movies;
  constructor() { }
  ngOnInit() {
  }
}
```

- กลับไปไฟล์ movie-list.component.html แล้วเพิ่มโค้ดแสดงผลรายชื่อหนังพร้อมข้อมูล

```
<ol>
  <li *ngFor="let movie of ms">
    <b>{{movie.name}}</b> ({{movie.year}}) <i>{{movie.genre}}</i>
  </li>
</ol>
```

Remarks

- ทุกครั้งที่เราสร้าง Component ใหม่ stackBlitz จะทำการ Import Component นั้นใน Module หลักให้อัตโนมัติ หากเราเขียนแอปเองจะต้องเพิ่มการ Import เอง
- ตอนเรา Import ข้อมูลรายชื่อหนังเข้ามา เราเอาข้อมูลมาใส่ในตัวแปรชนิด Array ชื่อเดียวกันกับที่กำหนดไว้ในไฟล์ข้อมูล แต่ Array ที่ได้จะเป็นตัวแปรแบบ hidden ไม่สามารถนำไปใช้ภายนอกได้จึงต้องมีการ export ไปยังตัวแปรใหม่ (ในที่นี้คือชื่อ ms) เราเรียกวิธีการนี้ว่า Binding

- *ngFor เป็น Directives ใน BrowserModule ดังนั้นเราต้อง Import Module นี้ถ้าต้องการใช้งาน โดย *ngFor จะทำงานเหมือน For Loop ของภาษาอื่น ๆ นั่นคือจะแสดงผล Tag นั้นซ้ำ ๆ กันตามจำนวนข้อมูลใน Array โดยมี Syntax ภายในเป็น

let item-name of array-name

Practice

- เพิ่มชื่อหนังอีกสามเรื่องในไฟล์ movies.ts (ไม่ซ้ำกัน สามารถแก้ไขเรื่องแรกได้หากเราไม่ชอบ Titanic)
- สร้างไฟล์ข้อมูล actors.ts โดยใช้รูปแบบข้อมูลดังนี้ โดยให้มีจำนวนนักแสดงอย่างน้อย 8 คน แต่แต่ละคนต้องแสดงหนังหนึ่งเรื่องจากสี่เรื่องในไฟล์ movies.ts และจะต้องมีนักแสดงอย่างน้อยหนึ่งคนจากหนังแต่ละเรื่อง

firstName: 'ชื่อ',

lastName: 'นามสกุล',

gender: 'M' (ชาย) หรือ 'F' (หญิง)

- สร้าง Component ใหม่ขึ้นมา เพื่อแสดงรายชื่อนักแสดงทั้งหมด จากนั้นแสดงผลลัพธ์ของ Component นี้ต่อท้ายรายชื่อหนังที่สร้างมาไว้แล้ว (สามารถใช้ Tag แทน ได้)
- ที่ไฟล์นามสกุล .css ของ Component ที่เราเพิ่งสร้างใหม่ เพิ่มโค้ดต่อไปนี้

```
.M {  
  color: blue  
}
```

```
.F {  
  color: red  
}
```

- ใน Tag ของรายชื่อนักแสดงให้เพิ่ม Attribute ที่ชื่อว่า class และค่าของ Attribute เป็น gender ของนักแสดงนั้น ๆ (M หรือ F)
หากให้ค่าที่ถูกต้องข้อความจะเป็นสีฟ้าสำหรับนักแสดงชายและสีแดงสำหรับนักแสดงหญิง

Hello Angular!

Start editing to see some magic happen :)

Movies

1. **Titanic** (1997) *drama*
2. **Titanic** (1997) *drama*
3. **Titanic** (1997) *drama*
4. **Titanic** (1997) *drama*

Actors

- [M] DiCaprio, Leonardo
- [F] DiCaprio, Leonardo
- [M] DiCaprio, Leonardo
- [F] DiCaprio, Leonardo
- [M] DiCaprio, Leonardo
- [M] DiCaprio, Leonardo
- [F] DiCaprio, Leonardo
- [F] DiCaprio, Leonardo

ตัวอย่างการแสดงผล