

PrimeNg

การใช้ PrimeNg ซึ่งการใช้งานคล้ายกับ Bootstrap มาก หากเคยได้ใช้ Bootstrap การเรียนรู้ PrimeNg ก็คงไม่ยาก แต่ PrimeNg ทำมาเพื่อใช้กับ Angular โดยเฉพาะจึงสามารถใส่ค่าหรือรับค่าจาก Angular ได้ดีกว่า นอกจากนี้ยัง UI ที่มากกว่าโดยเฉพาะการทำกราฟในรูปแบบต่าง

เริ่มต้นกับ PrimeNg

เมื่อให้เห็นภาพการใช้งานพื้นฐานการใช้ PrimeNg ว่าขั้นตอนต้องเตรียมอะไรบ้าง ซึ่งลำดับต่อไปนี้จะเริ่มต้นติดตั้ง PrimeNg และทดสอบการใช้งานกับคอมโพเนนต์พื้นฐาน สิ่งแรกคือต้องติดตั้งดังนี้

```
ng new myAngular
cd myAngular
npm install primeng --save
npm install primeicons --save
npm install @angular/animation --save
```

ตรวจสอบว่า มีไฟล์เดอร์ priming และ primeicons ใน myAngular\node_modules แล้ว และตรวจสอบว่า package.json มี "primeicons": "^2.0.0", "primeng": "^9.0.6" ในส่วน **dependencies** ต้องอยู่ dependencies เท่านั้น ถ้าไปอยู่ใน ส่วนอื่นถือว่าผิด (เลขที่รุ่นขึ้นกับการติดตั้ง ณ เวลา นั้นว่าได้รุ่นใด) ณ ขณะนี้ยังไม่ได้เปิด myAngular ที่สร้างไว้ ต่อมาจึงต้องเปิด

```
ng serve --open
```

ใส่เพิ่ม CSS ของ primeNg ใน angular.json (ใส่ theme ชื่อ nova-light ซึ่งเปลี่ยนที่หลังได้)

Code 1. angular.json

```
"styles": [
  "node_modules/primeicons/primeicons.css",
  "node_modules/primeng/resources/themes/nova-light/theme.css",
  "node_modules/primeng/resources/primeng.min.css",
]
```

เริ่มทดสอบคอมโพเนนต์ primeNg ด้วย Button เพิ่มในไฟล์ app.module.ts และใส่เพิ่ม ButtonModule และ BrowserModule ในส่วน import

Code 2. src/app/app.module.ts

```
import {ButtonModule} from 'primeng/button';
import {BrowserAnimationsModule} from '@angular/platform-browser/animations';

@NgModule({
  imports: [
    BrowserModule,
    AppRoutingModule,
    ButtonModule,
    BrowserAnimationsModule,
  ],
})
```

```
//..
})
export class AppModule { }
```

ต่อมาใส่ p-button และ pi-spinner ในส่วน app.component.html (กรณีนี้ทดสอบในหน้าหลัก)

Code 3. src/app/app.component.html

```
<h3 class="first">Button Component</h3>
<p-button label="Click"></p-button>
<p-button icon="pi pi-check" label="Click"></p-button>
<p-button icon="pi pi-check" iconPos="right" label="Click"></p-button>
<p-button icon="pi pi-check"></p-button>
<p-button icon="pi pi-check" [disabled]="true" label="Disabled"></p-button>
<br>
<i class="pi pi-spin pi-spinner" style="font-size: 3em"></i>
```

การนำเข้า อะไรที่เป็นโมดูล (สังเกตว่ามีคำว่า module ต่อท้าย) จะต้องดำเนินการที่ไฟล์ app.module.ts ดังตัวอย่างการใช้ ButtonModule แต่ไม่ใช่ทุกตัวใน PrimeNg จะเป็นโมดูล เช่น MenuItem ซึ่งมาจาก primeng/api ตัวนี้ไม่ใช่เป็นโมดูลเป็นเพียงไทป์ (type) หนึ่ง จึงไม่ต้องดำเนินการแบบเดียวกับโมดูล แต่ให้นำเข้า import ในคอมโพเนนท์ ที่เรียกใช้ได้เลย เช่น

```
import {MenuItem} from 'primeng/api';
```

ในการนำไปใช้กับคอมโพเนนท์ ต้องการไทป์นี้ในการอ้างอิงกับคอมโพเนนท์ เช่น

```
items: MenuItem[ ];
```

ผลที่ได้ จากใช้ ButtonModule กับอิลิเมนต์ <p-button> และทดสอบการเคลื่อนไหวกับ pi-spinner จะได้ผลตามรูปนี้



รูป 1 ผลการทดสอบของ PrimeNG

ได้ผลดังรูปนี้ถือว่าการใช้งาน PrimeNg ทำงานได้สมบูรณ์แล้ว ซึ่งก็พร้อมจะศึกษาในการใช้งาน PrimeNg กันต่อไป

MenuModule

โมดูลแรกที่จะแนะนำให้ฝึกใช้คือ MenuModule โมดูลนี้จะ เมนูอื่นๆ ให้ใช้ได้หลายตัว เช่น Breadcrumb, ContextMenu, MegaMenu, MenuBar ซึ่งอยากใหลองนึกถึง Bootstrap ที่ก็มีเมนูต่างๆ ให้เลือกใช้มากมาย

สิ่งแรกคือ ให้นำเข้า MenuModule ก่อน เนื่องจากเป็นโมดูลแล้วยังมีไทป์อื่นๆ ที่ใช้กับโมดูลนี้ ดังนั้นเมื่อต้องนำเข้าอะไรที่เกี่ยวกับโมดูลนี้ก็ต้อนำเข้า เช่นถ้าต้องการใช้ ไทป์ MenuItem ก็ต้อนำเข้าไทป์นี้ด้วย

Code 4. src/app/app.module.ts

```
import {MenubarModule} from 'primeng/menubar';
@NgModule({
  imports: [
    BrowserModule,
    AppRoutingModule,
    ButtonModule,
    BrowserAnimationsModule,
    MenubarModule,
  ],
  //..
})
```

สำหรับคอมโพเนนท์ ต้องนำเข้า MenuItem ในการเรียกใช้ต้องประกาศตัวแปร สมมติให้ชื่อ items และมีไทป์เป็น MenuItem[] ซึ่งเป็นอาร์เรย์ แต่ละรายการประกอบด้วยชื่อ แทนด้วย label และถ้าต้องการมีรายการย่อยก็ประกาศต่อเนื่องได้ในรายการนี้ นอกจากนี้ยังสามารถใส่รูปไอคอน ซึ่งแทนด้วย icon ได้ ดังตัวอย่างต่อไปนี้

Code 5. src/app/app.component.ts

```
import { Component } from '@angular/core';
import {MenuItem} from 'primeng/api';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'myAngular';
  items: MenuItem[];
  constructor() { }

  ngOnInit(): void {
    this.items = [{
      label: 'File',
      items: [
        {label: 'New', icon: 'pi pi-plus'},
        {label: 'Open', icon: 'pi pi-download'}
      ]
    },
    {
      label: 'Edit',
      items: [
        {label: 'Undo', icon: 'pi pi-refresh'},
        {label: 'Redo', icon: 'pi pi-repeat'}
      ]
    }
  ]};
}
```

จากตัวอย่างนี้จะเห็นว่า มีสองรายการ รายการแรกมีเมนูย่อยชื่อ New และ Open และรายการที่สองมีรายการย่อยชื่อ Undo และ Redo

ต่อไปจะเป็นการนำเมนูไปแสดงหน้าเว็บ ด้วยชื่ออีลิเมนต์ <p-menubar> โดยมีการผูกข้อมูลกับ ตัวแปร items ดังที่เขียนในคลาสของคอมโพเน้นท์นี้ นอกจากนี้ยังสามารถใส่ <input> และ <button> เพิ่มเข้า <p-menubar> ได้อีก

Code 6. src/app/app.component.html

```
<p-menubar [model]="items">
  <div>
    <input type="text" pInputText placeholder="Search">
    <button pButton label="Logout" icon="pi pi-power-off"
      style="margin-left:.25em"></button>
  </div>
</p-menubar>
```



รูป 2 ผลการทดสอบของ MenuItem