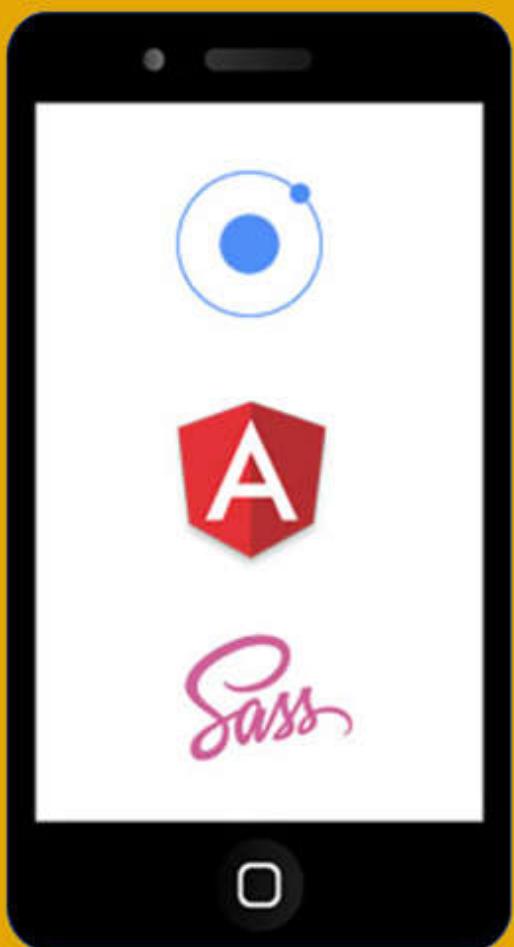


e-Book

# สร้าง Mobile App ด้วย Ionic 3 และ Angular



โค๊ชเอก

# Build Mobile App with Ionic 3 and Angular

สร้าง Mobile App ด้วย Ionic 3 และ Angular

โค้ชเอก

หนังสือเล่มนี้จำหน่ายที่ <https://codingthailand.com/ionic3angularebook>

เวอร์ชัน 3 ออกจำหน่ายวันที่ 1 สิงหาคม 2561

หนังสือเล่มนี้ผู้เขียนตั้งใจจัดทำขึ้นเพื่ออย่างให้มีหนังสือภาษาไทยทั้งเล่มเกี่ยวกับ Ionic Framework 3 ที่เน้นเนื้อหาตั้งแต่พื้นฐาน และเน้นให้ผู้อ่านได้มีความเข้าใจถึงกระบวนการพัฒนา Mobile App เครื่องมือต่างๆ รวมถึงภาษาโปรแกรม ไลบรารี ไม่ว่าจะเป็น Node.js, Angular, TypeScript และ SASS เป็นต้น นอกจากความรู้สำหรับการพัฒนา Mobile App แล้ว ผู้อ่านจะได้เรียนรู้เทคโนโลยีใหม่ๆควบคู่กันไปด้วย หวังว่าหนังสือเล่มนี้จะเป็นประโยชน์ และประทับใจในการเรียนรู้ให้กับทุกคนครับ

“จะเขียนนวนิยายไม่รู้ ด้วยการพัฒนาตัวเอง และลงมือทำอย่างสม่ำเสมอ”



©2019 โค้ชเอก

## สารบัญ

### บทที่ 1 การติดตั้ง Node.js, npm และ Visual Studio Code 8

- การติดตั้ง Node.js และ npm
- การตรวจสอบเบอร์ชัน Node.js และ npm
- รีจิกกับ npm
- สรุปคำสั่ง npm พื้นฐาน
- การติดตั้ง Visual Studio Code
- การติดตั้ง Extension ให้กับ Visual Studio Code

### บทที่ 2 พื้นฐาน Angular , TypeScript และ Sass สำหรับ Ionic 15

- พื้นฐาน Angular
- Angular Modules
- Angular Decorators
- Angular Components
- พื้นฐาน TypeScript
- ชนิดข้อมูลของ TypeScript และการประกาศตัวแปร
- พื้นฐาน Sass (Syntactically Awesome StyleSheets)

### บทที่ 3 การติดตั้ง Ionic 19

- การติดตั้ง Ionic Framework
- การสร้างโปรเจคใหม่
- ลงมือสร้างโปรเจคใหม่
- เพิ่ม Platforms ที่เราต้องการ Build
- การสั่งรัน Application ไปที่ Browser
- การสั่งรัน Application ไปที่ เครื่องทดสอบจริง
- การสั่งรัน Application ไปที่ Emulator
- การอัปเดต Ionic CLI ให้เป็นเวอร์ชันล่าสุด
- การเปิดโปรเจค และเพิ่มเข้าใน Visual Studio Code

## บทที่ 4 แนะนำ Ionic Framework 26

- โครงสร้างไฟล์ และไฟล์เดอร์ต่างๆ ในโปรเจค
- การใช้ IONIC CLI COMMANDS เพื่อสร้างโค้ดอัตโนมัติ
- การใช้ Command line ภายใต้โปรแกรม Visual Studio Code

## บทที่ 5 การสร้าง Pages, Basic Navigation และ Passing Data 29

- การสร้าง Pages
- ทดลองสร้างเพจชื่อว่า About
- ทดลองสร้างเมนู “เกี่ยวกับเรา” เพื่อเปิดหน้าเพจ About
- พื้นฐาน Navigation
- การ Push หน้าเพจขึ้นมาแสดง
- การ Pop หน้าเพจ
- การกำหนดเพจ ให้เป็น root page
- การส่งข้อมูลระหว่างเพจ (Passing Data between Pages)

## บทที่ 6 Ionic Components 41

- Components คืออะไร
- การใช้ Components เป็นต้น

## บทที่ 7 การเรียกใช้ API ด้วย HTTP Services 44

- ขั้นตอนการใช้ HTTP Services หรือ provider ใน Ionic
- การสร้าง Pages ในรูปแบบ Master Detail
- การใช้ Component Search bar สำหรับค้นหาข้อมูล

## บทที่ 8 การสร้างฟอร์ม (Form) 72

- ลองสร้างฟอร์มสมัครสมาชิก (ลงทะเบียน)
- การตรวจสอบความถูกต้องของข้อมูล (Validations)
- การเพิ่มข้อมูลไปที่ Backend

<b>บทที่ 9 การเก็บข้อมูลแบบ Local ด้วย Ionic Storage</b>	<b>81</b>
● การติดตั้ง Storage	
● ขั้นตอนการใช้งาน และเก็บข้อมูลแบบ Key/Value	
● การใช้งาน Ionic Storage เพื่อเก็บข้อมูลแบบฐานข้อมูล (SQLite) (แสดง,เพิ่ม,แก้ไข,ลบข้อมูล)	
<b>บทที่ 10 การเขียน Ionic เชื่อมต่อกับ Firebase</b>	<b>93</b>
● คุณสมบัติของ Firebase	
● ลองใช้ Realtime Database (แสดง,เพิ่ม,แก้ไข,ลบข้อมูล)	
<b>บทที่ 11 การใช้งาน Ionic Native</b>	<b>101</b>
● การใช้งานกล้อง (Camera)	
● การใช้งาน Barcode Scanner และ QR Code	
<b>บทที่ 12 การตกแต่ง App ด้วย Sass และ จัดรูปแบบการแสดงข้อมูลด้วย Pipes</b>	<b>109</b>
● การทำ Theming ให้กับ App	
● Utility Attributes	
● การตกแต่ง App โดยการเขียนทับตัวแปร Sass ของ Ionic	
● จัดรูปแบบการแสดงข้อมูลด้วย Pipes	
● พื้นฐานการใช้งาน Built-in pipes ใน Angular	
● การสร้าง Custom Pipes ใน Ionic	
<b>บทที่ 13 การเตรียม Resources และ Publish App</b>	<b>118</b>
● ตั้งค่า Bundle ID, App Name, เลขเวอร์ชัน ให้กับ App	
● การลดขนาดรูปภาพก่อน Build App	
● การสร้างไอคอน, การ Publish App ทั้ง Android และ iOS (ศึกษาผ่านวิดีโอ)	

## บทที่ 14 ใบนัสพิเศษ

119

- ตัวอย่างไฟล์ psd (PhotoShop) สำหรับทำ splash screen
- ไฟล์ตั้งค่าทั้งหมดในหนังสือ

### Changelog

เวอร์ชัน 3 – 1 สิงหาคม 2561

- Ionic CLI 3.20
- Ionic Framework 3.9.2
- Angular 5

เวอร์ชัน 1 – 1 มีนาคม 2561

- Ionic CLI 3.19.1
- Ionic Framework 3.9.2
- Angular 5

Ionic Framework 3 เป็น Mobile Framework สมัยใหม่ที่ช่วยให้เราพัฒนา Mobile App ได้อย่างรวดเร็ว และง่ายมากๆ แนวคิดคือ การพัฒนา Mobile App ให้เหมือนกับพัฒนา Web App โดยที่นักพัฒนาแต่ละคนจะต้องมีความรู้พื้นฐาน HTML, CSS และ JavaScript (JS ES6) มาแล้วถึงจะสามารถเรียนรู้ Ionic ได้อย่างรวดเร็ว

Ionic Framework ในเวอร์ชัน 3 นี้ ข้อดีของผู้เริ่มต้นคือ ไม่จำเป็นต้องรู้ Ionic เวอร์ชัน 1 มาก่อนก็สามารถเรียนรู้ได้ทันที เพราะตัว Ionic 3 เอง ได้ถูกปรับปรุงให้เข้ากับ Angular 5 และ TypeScript เพื่อให้ระบบสนับสนุนการเขียนโค้ดแบบ TypeScript มากขึ้น ทำให้การสร้าง Mobile App ที่ซับซ้อนมากขึ้น ก็ข้อแนะนำว่าให้ศึกษา Angular ด้วย มาลุยกันเลย!

## บทที่ 1 การติดตั้ง Node.js, npm และ Visual Studio Code

ก่อนเขียน Ionic เราต้องติดตั้งซอฟต์แวร์ และเครื่องมือกันก่อน ประกอบด้วย Node.js, npm และ Visual Studio Code

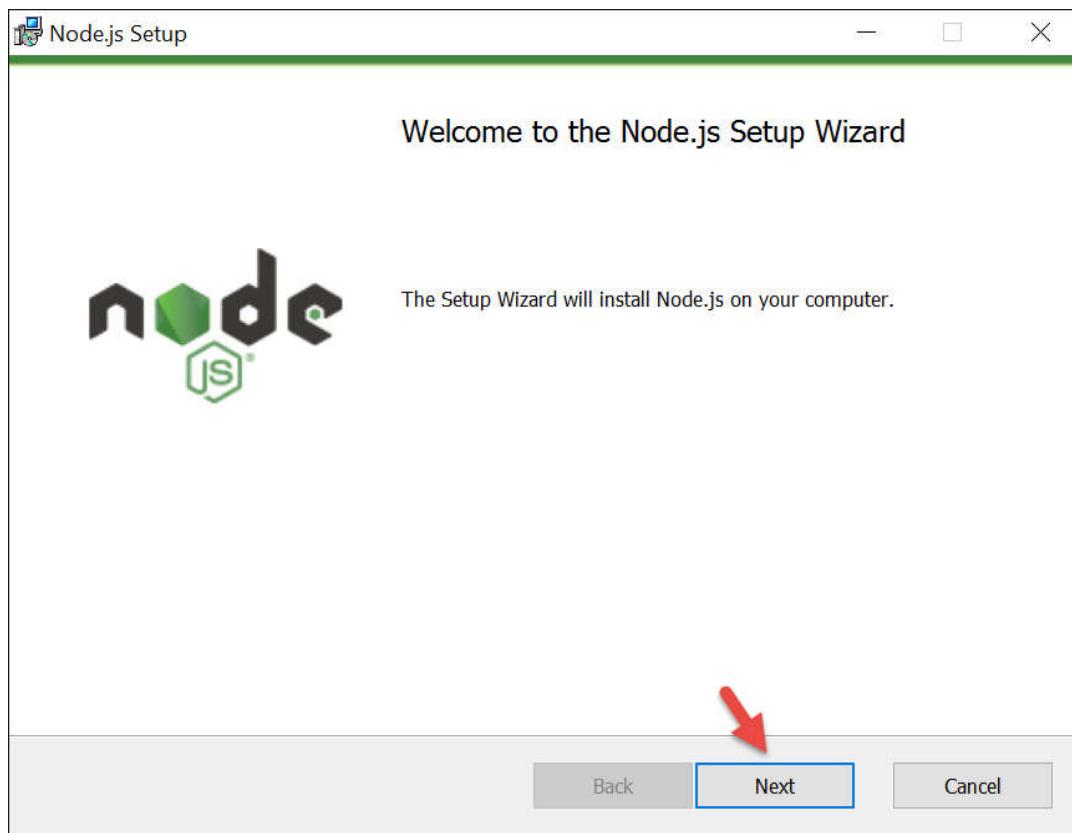
### การติดตั้ง Node.js และ npm

การติดตั้ง Node.js สามารถดาวน์โหลดการติดตั้งได้โดยตรงที่เว็บ <https://nodejs.org/en/download> เราชำนารถติดตั้งได้ทั้ง Windows, Linux และ Mac ในหนังสือเล่มนี้จะใช้ Node.js สำหรับ Windows เป็นหลัก แต่ถ้าใครใช้ Mac หรือ Linux ก็ใช้ได้ไม่มีปัญหา เพราะคำสั่งต่างๆ นั้นเหมือนกัน

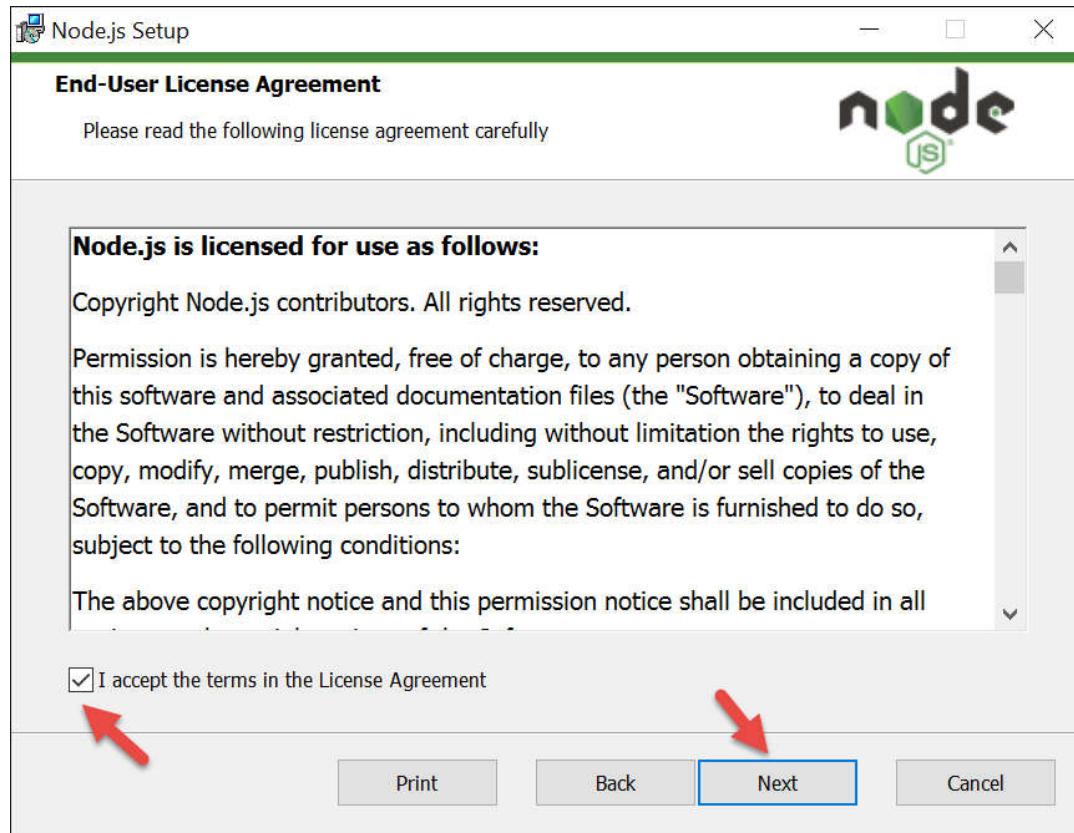
ในการเขียน Ionic Framework 3 แนะนำให้ใช้ Node.js เวอร์ชัน **LTS** (สำหรับเครื่อง mac หากมีปัญหาในการใช้งานสามารถอยู่ลงมา 1 เวอร์ชันก็ได้เช่นเดียวกัน)

หลังจากดาวน์โหลดมาแล้วให้ติดตั้งตามขั้นตอน ดังนี้

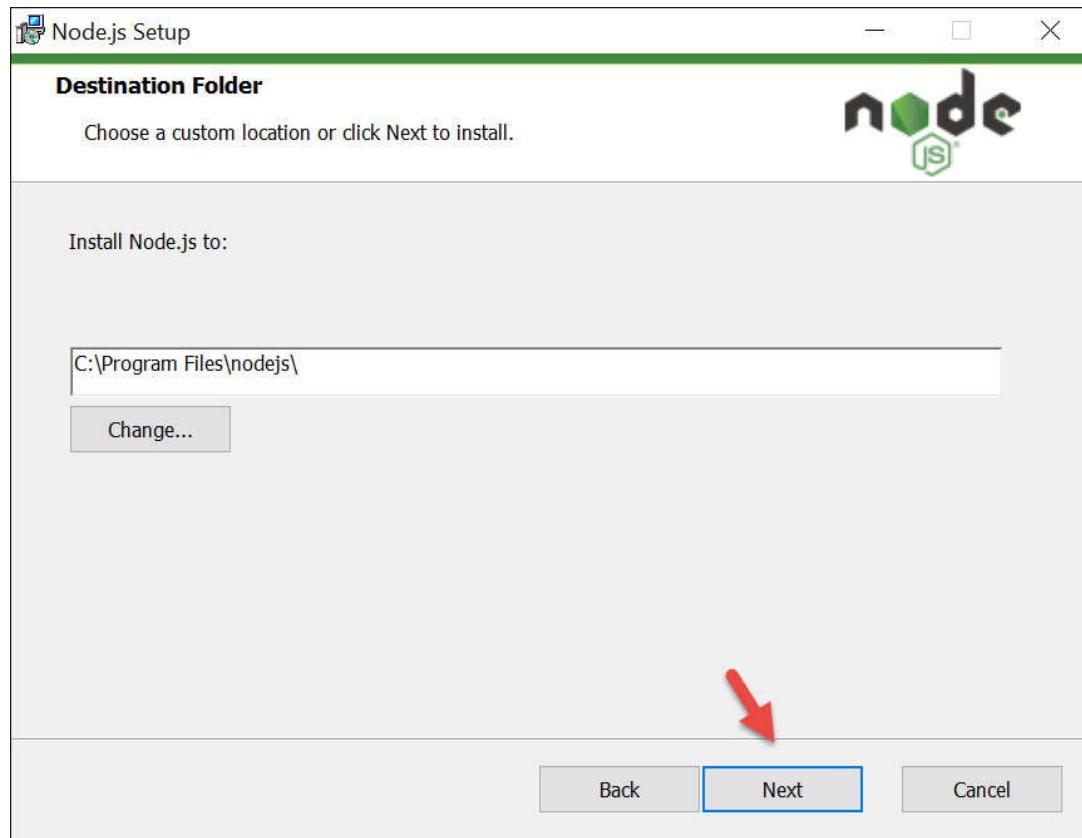
1. ดับเบิลคลิกไฟล์ติดตั้ง แล้วกดปุ่ม Next



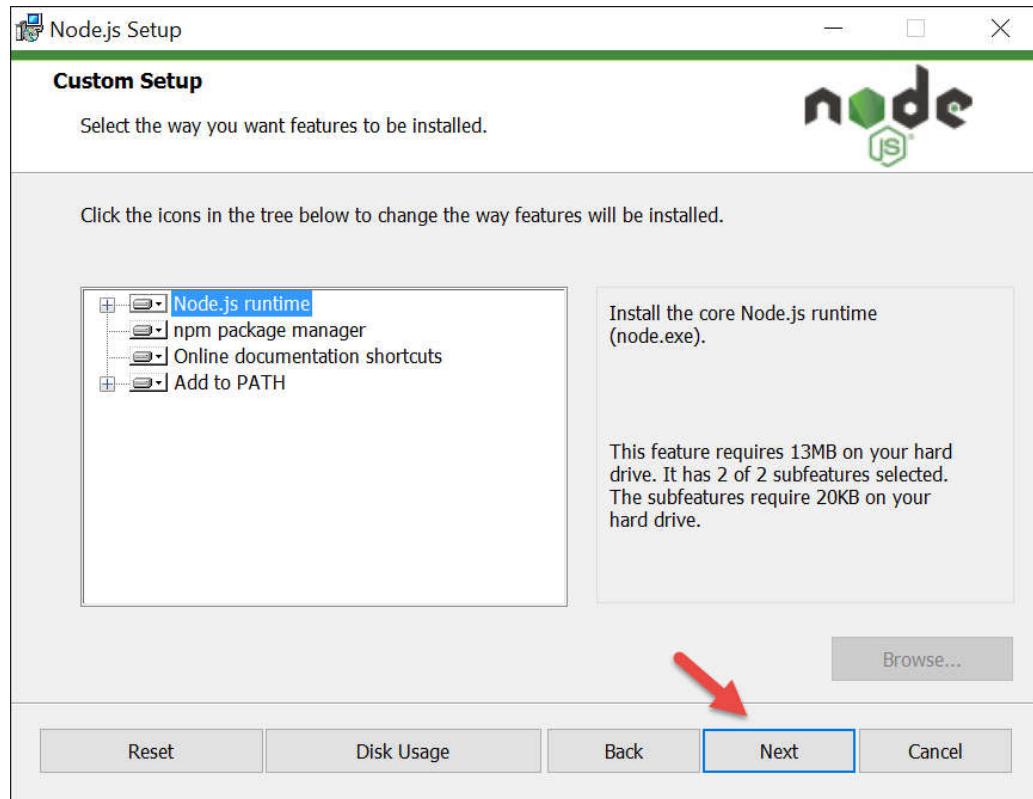
2. ດັກເລືອກ I accept the terms... ແລ້ວກົດປຸ່ມ Next



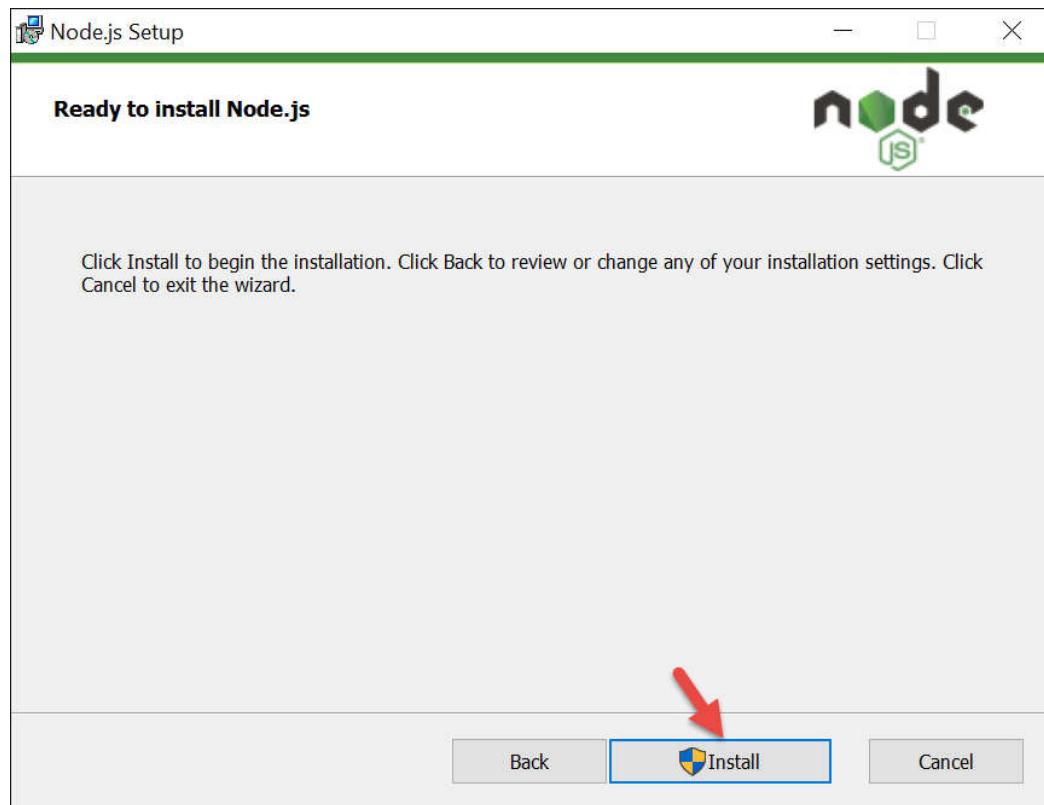
3. ກົດປຸ່ມ Next



4. กดปุ่ม Next (แนะนำว่าไม่ต้องแก้อะไร)

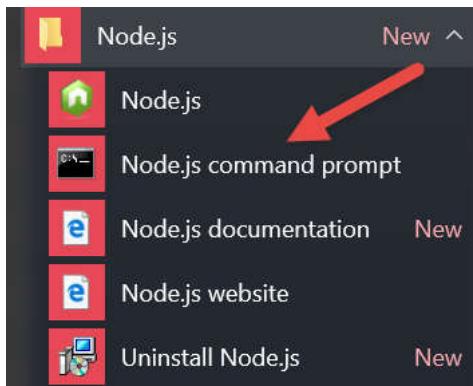


5. คลิกปุ่ม Install เพื่อติดตั้ง Node.js



## การตรวจสอบเวอร์ชัน Node.js และ npm

หลังจากติดตั้ง Node.js เรียบร้อย ให้ตรวจสอบเวอร์ชันของ Node.js และ npm ด้วย เพื่อตรวจสอบว่า เราได้ติดตั้ง Node.js สมบูรณ์พร้อมใช้งานหรือไม่ ให้เปิด Node.js command prompt (จะค้นหา หรือคลิกที่ Start Menu ของ Windows ก็ได้)



เมื่อเปิดขึ้นมาแล้วให้พิมพ์คำสั่ง node -v และกด enter เพื่อตรวจสอบเวอร์ชันของ Node.js และพิมพ์ npm -v และกด Enter เพื่อตรวจสอบเวอร์ชันของ npm ดังรูป

Command Prompt

```
C:\>node -v
v8.9.4
```

```
C:\>npm -v
5.6.0
```

Note: การเปิดใช้งาน Node.js command prompt ทุกครั้ง แนะนำให้คลิกขวา แล้วเลือก Run as administrator ทุกครั้ง

## รู้จักกับ npm

npm เป็นตัวจัดการ dependencies หรือไลบรารี ต่างๆ ในโปรเจคของเรา ข้อดีคือ เราไม่ต้องเข้าเว็บเพื่อไปดาวน์โหลดไฟล์มาติดตั้งที่ลําตัว เราสามารถใช้ npm จัดการการติดตั้ง หรือลบ dependencies ต่างๆ ได้อย่างสะดวก (ผ่านอินเทอร์เน็ต)

### สรุปคำสั่ง npm พื้นฐาน

- npm init เริ่มต้นให้งาน และช่วยให้เราสร้าง และใส่รายละเอียดเกี่ยวกับโปรเจคของเรา (package.json)
- npm install module\_name ติดตั้ง module
- npm install -g module\_name ติดตั้ง module ในระดับ global
- npm install module\_name --save ติดตั้ง module และเพิ่มมันเข้าไปในไฟล์ package.json (ข้างใน dependencies)
- npm install module\_name --save-dev เหมือนกันกับข้างบน ติดตั้ง module และเพิ่มมันเข้าไปในไฟล์ package.json (ข้างใน dependencies) และบอกว่าจะนำมาช่วยในส่วนของการพัฒนา
- npm list แสดงรายการ modules ทั้งหมดที่ติดตั้งไปแล้วในโปรเจคของเรา
- npm list -g แสดงรายการ modules ระดับ global ทั้งหมดที่ถูกติดตั้งบน OS ของเรา
- npm uninstall module\_name uninstall module ออกจากโปรเจคของเรา
- npm uninstall -g module\_name uninstall module ระดับ global
- npm uninstall module\_name --save uninstall module ออกจากโปรเจคของเรา พิริมทั้งหมดออกจาก attribute dependencies ในไฟล์ package.json
- npm uninstall module\_name --save-dev เหมือนกันกับด้านบน uninstall module ออกจากโปรเจคของเรา พิริมทั้งหมดออกจาก attribute dependencies ในไฟล์ package.json
- npm update module\_name update เออร์ชัน module ให้ใหม่ล่าสุด
- npm update -g module\_name update เออร์ชัน module ให้ใหม่ล่าสุด ในระดับ global
- npm -v แสดงเวอร์ชันปัจจุบันของ npm

Note: เว็บไซต์หลักของ npm <https://www.npmjs.com/>

## การติดตั้ง Visual Studio Code

Visual Studio Code เป็น Editor สมัยใหม่ โดยมีคุณสมบัติเด่น คือ สามารถใช้งานได้พร้อมๆ กัน Cross-platform และมีตัวจัดการ Extensions หรือส่วนเสริมต่างๆ เพื่อช่วยให้นักพัฒนาเขียนโค้ดได้ง่ายขึ้น

Note: เว็บไซต์หลักของ VS Code <https://code.visualstudio.com/>

เนื่องจากการติดตั้ง VS Code ค่อนข้างง่ายเพียงแค่กดปุ่ม next ก็เสร็จเรียบร้อยจึงไม่ขอกล่าวถึงในหนังสือเล่มนี้แต่ถ้าหากสนใจสามารถมาได้

### การติดตั้ง Extensions ให้กับ Visual Studio Code

สำหรับการเขียน Ionic มี Extensions ที่แนะนำให้ติดตั้ง ดังนี้

**Angular Extension Pack** ใช้สำหรับเขียน Angular ได้ง่ายขึ้น โดยมี extensions หลายตัวอยู่ในนี้

<https://marketplace.visualstudio.com/items?itemName=doggy8088.angular-extension-pack>

**vscode-icons**

ใช้สำหรับแสดงไอคอนไฟล์ต่างๆ ในโปรเจคให้ดูง่าย และสวยงาม

<https://marketplace.visualstudio.com/items?itemName=robertohuertasm.vscode-icons>

**VS Color Picker**

ใช้สำหรับเลือกสีให้กับ CSS

<https://marketplace.visualstudio.com/items?itemName=lihui.vs-color-picker>

**Sass**

ใช้สำหรับเขียน Sass ให้ง่ายขึ้น

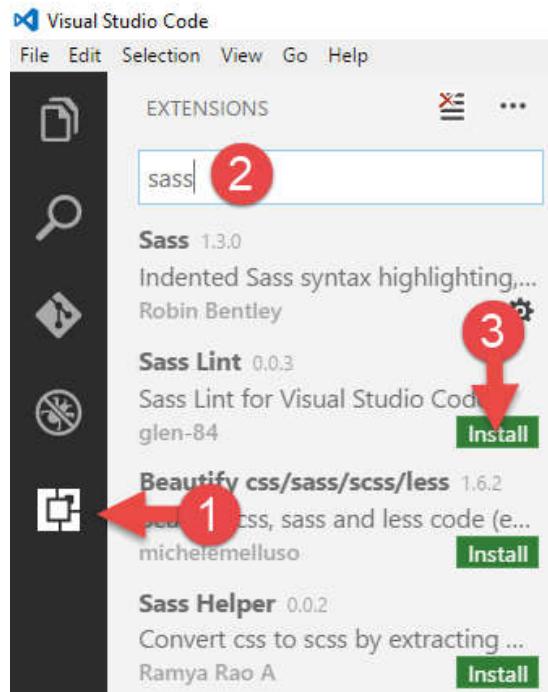
<https://marketplace.visualstudio.com/items?itemName=robinbentley.sass-indented>

ขั้นตอนการติดตั้ง Extension ต่างๆ มีดังนี้

1. เปิด VS Code ขึ้นมา คลิกเลือกที่ไอคอน Extensions



2. พิมพ์ชื่อ Extension ที่ต้องการ แล้วคลิกปุ่ม install



3. เมื่อกดปุ่ม install เรียบร้อย อาจต้องคลิก reload เพื่อเปิด VS code อีกครั้ง จึงจะสามารถใช้งานได้

## บทที่ 2 พื้นฐาน Angular, TypeScript และ Sass สำหรับ Ionic

### พื้นฐาน Angular

Angular เป็น JavaScript framework หรือเรียกได้ว่าเป็น platform สำหรับสร้าง applications ผ่าน client เขียนด้วย TypeScript ในการพัฒนา Ionic 3 นั้น เราจะต้องเขียนโค้ดด้วย Angular (TypeScript) ดังนั้นจึงจำเป็นอย่างมากที่เราต้องเรียนรู้ Angular ด้วย

#### Angular Modules

Module เป็นแนวคิดในการเขียนโค้ดชุดหนึ่งขึ้นมาเพื่อใช้งานด้วยจุดประสงค์บางอย่าง เราสามารถ import module เข้ามาใช้งานได้ตลอดเวลา จะเป็น Module ที่เราเขียนขึ้นมาเอง หรือจะ import library modules ของ Angular เข้ามาใช้ก็ได้ ยกตัวอย่าง เช่น

```
import { Component } from '@angular/core';
import { MyDetailsPage } from '../my-details/my-details';
```

```
@Component({
  templateUrl: 'home.html'
})
```

```
export class HomePage {
  constructor(){}
}
```

จากตัวอย่างโค้ดเป็นนำเข้า (import) Angular Component function จาก module @angular/core และถ้าเราต้องการส่งออก (export) พังก์ชันหรือคลาส ก็สามารถเขียนคำสั่ง export ไว้ด้านหน้าได้เลย

## สรุปเรื่อง Modules

1. App ที่เขียนด้วย Angular จะประกอบขึ้นมาด้วย Modules ต่างๆ ซึ่งทำหน้าที่แตกต่างกัน
2. Modules จะ export บางอย่าง ได้แก่ classes, function, values แต่ละ modules อาจมีการ import modules ต่างๆ เข้ามาใช้งานได้
3. การเขียน Application ด้วย Angular ก็คือ การเขียนเพื่อเรียกใช้ Modules ต่างๆ นั่นเอง

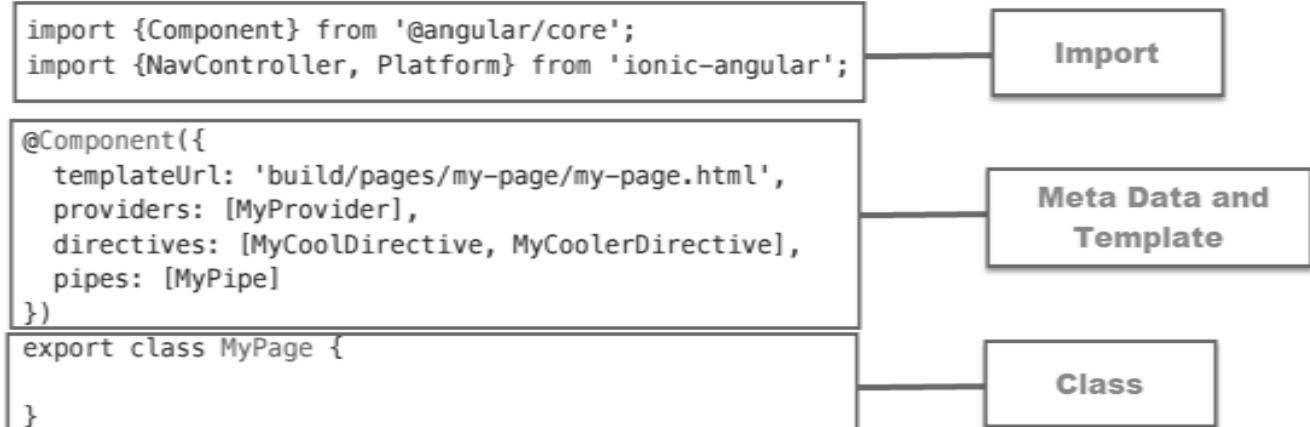
Note: หลักการ import และ export นี้ เป็นหลักการของ JavaScript ES2015 หรือ ES6

## Angular Decorators

Decorators เป็นฟังก์ชันที่ความสามารถเพิ่ม meta data ต่างๆ ให้กับ Class อาจเป็นสมาชิกของคลาสหรือ method ของคลาสก็ได้ Decorators จะเขียนด้วยเครื่องหมาย @ เช่น @Component, @Pipe เป็นต้น

## Angular Components

ใน Angular นั้น Component คือ ส่วนที่ใช้ในการควบคุมการแสดงผล หรือเรียกว่าง่ายๆ ว่า ส่วน view ประกอบด้วย Template, Class (properties + methods) และ meta data ที่เกี่ยวข้องกับการแสดงผล และ logic ต่างๆ ที่ใช้ในหน้าเพจ ส่วนประกอบ Components มีดังนี้



Note: ใน Ionic หากเปลี่ยนเที่ยงง่ายๆ 1 Component ก็คือ 1 หน้า (Page) ของ Mobile App นั่นเอง

## พื้นฐาน TypeScript

TypeScript เป็นส่วนขยาย และเป็น superset ของภาษา JavaScript โดยเราสามารถเขียนระบุชนิดข้อมูล หรือ type ได้ สามารถเขียน Classes, Interfaces หรือ Generics ได้ (คล้ายกับการเขียนภาษา C# หรือ java) แนะนำใน Ionic ใช้ TypeScript ในการเขียนได้เป็นหลัก

Note: หากอยากรู้ TypeScript เพิ่มเติม แนะนำเว็บ <https://www.typescriptlang.org/docs/handbook/basic-types.html>

### ชนิดข้อมูลของ TypeScript และการประกาศตัวแปร

ชนิดข้อมูลของ TypeScript คล้ายกับ JavaScript ประกอบชนิดข้อมูลที่ใช้บ่อย ดังนี้

1. Boolean เป็นข้อมูลชนิดเก็บค่า true หรือ false

```
let isDone: boolean = false;
```

2. Number เป็นชนิดข้อมูลเก็บค่าตัวเลข โดยปกติแล้ว Number จะเป็น floating point

```
let decimal: number = 6;
```

```
let hex: number = 0xf00d;
```

```
let binary: number = 0b1010;
```

```
let octal: number = 0o744;
```

3. String เป็นชนิดข้อมูลเก็บข้อความต่างๆ โดยจะอยู่ในเครื่องหมาย quotes ("") หรือ single quotes ('')

```
let color: string = "red";
```

```
let fullName: string = 'Bob Bobbington';
```

```
let age: number = 37;
```

```
let sentence: string = `Hello, my name is ${ fullName }.
```

```
I'll be ${age + 1} years old next month.'
```

สามารถใช้ Template strings ได้โดยใช้เครื่องหมาย backtick (`) ครอบ string ที่ต้องการ หากต้องการแสดงตัวแปรให้ใช้ \${ }

4. Array เป็นชนิดข้อมูลแบบชุดสามารถเก็บค่าได้หลายค่าภายในตัวแปรเดียว

```
let list: number[] = [1, 2, 3];
```

```
let list: Array<number>;
```

```
let items: Array<{}>;
```

5. Any เป็นชนิดข้อมูลที่ประกาศไว้หากยังไม่รู้ว่าจะเก็บค่าชนิดข้อมูลเป็นอะไร หรือจะเรียกว่าการเก็บข้อมูลแบบ dynamic ก็ได้

```
let notSure: any = 4;
```

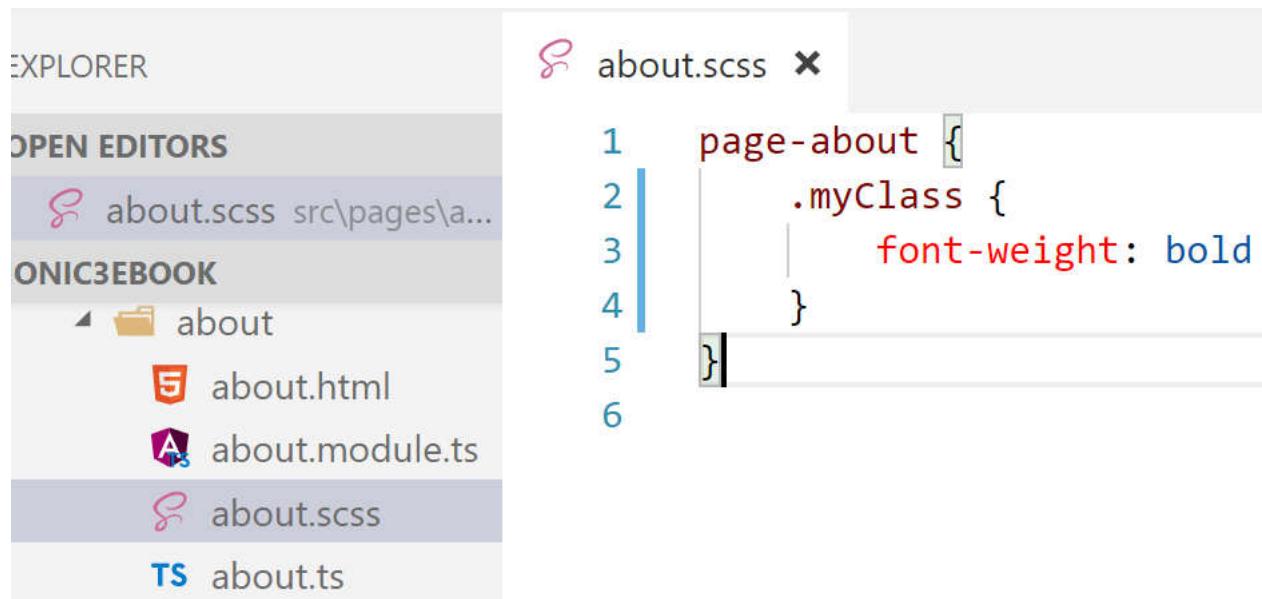
```
let notSure: any = null;
```

## พื้นฐาน Sass (Syntactically Awesome StyleSheets)

Sass เป็น stylesheet language (จะถูก complies เป็น CSS อีกที) ใช้สำหรับตกแต่งหน้าเพจให้สวยงาม โดยมีความสามารถ เช่น variables, nesting และ mixins เป็นต้น ใน Ionic เราสามารถเขียน Sass หรือจะเขียน CSS ปกติได้ เช่นเดียวกัน การเขียน **SASS** นั้นสามารถเปลี่ยนได้ในไฟล์ **.SCSS**

Note: หากต้องการศึกษาข้อมูลเพิ่มเติมดูได้ที่เว็บ <http://sass-lang.com/>

ตัวอย่างการเขียน เราสามารถเขียนและอ่าน CSS ปกติได้ เช่นเดียวกัน ดังนี้



```

EXPLORER
OPEN EDITORS
  about.scss src\pages\about
IONIC3EBOOK
  about
    about.html
    about.module.ts
  about.scss
  about.ts

about.scss
1 page-about {
2   .myClass {
3     font-weight: bold
4   }
5 }
6

```

## บทที่ 3 การติดตั้ง Ionic

### การติดตั้ง Ionic Framework

การติดตั้ง Ionic Framework มีขั้นตอน ดังนี้

- เปิด Node.js command prompt และพิมพ์คำสั่ง `cd C:\` แล้วกด Enter (จะสร้างโปรเจคใหม่ที่ Drive C สามารถเปลี่ยนได้ตามความต้องการ สำหรับคนที่ใช้ Mac สามารถเลือกโฟลเดอร์ที่ต้องการติดตั้งได้เลย)

```
C:\Windows\System32>cd C:\  
C:\>
```

**Note:** ก่อนเปิด Node.js command prompt อย่าลืม *Run as administrator*

- ติดตั้ง Ionic พิมพ์คำสั่ง `npm install -g ionic@3` แล้วกด Enter รอกจนเสร็จเรียบร้อย

**Note:** หากใช้ MAC หรือ Linux ให้ใช้คำสั่ง `sudo npm install -g ionic`

- ติดตั้ง Cordova ให้ใช้คำสั่ง `npm install -g cordova` แล้วกด Enter รอกจนเสร็จเรียบร้อย

**Note:** หากใช้ MAC หรือ Linux ให้ใช้คำสั่ง `sudo npm install -g cordova`

- ติดตั้ง Android Studio และ Android SDK ดาวน์โหลดที่ <https://developer.android.com/studio/index.html>

**Note:** สำหรับคนที่ใช้ MAC หากต้องการ build ios อย่าลืมติดตั้ง XCode ด้วย

**Note:** แนะนำให้ติดตั้ง JAVA JDK (เวอร์ชัน 8) ด้วยที่ <http://bit.ly/2LChu69>

**Note:** แนะนำให้ติดตั้ง git ให้ติดตั้ง Ionic 3 แบบ lokale ให้ความเข้าใจที่ดีขึ้น ได้ที่ <http://bit.ly/2LQXYSW>

- ติดตั้ง git scm ได้ที่ <https://git-scm.com> ให้คลิกดาวน์โหลดไฟล์ และติดตั้งตามปกติ (ไม่ต้องแก้-obr) หลังจากติดตั้ง git เรียบร้อยให้ตั้งค่า ชื่อและอีเมลของเราด้วย โดยเปิด command prompt ขึ้นมาแล้วรันสองคำสั่ง ดังรูป

### █ Command Prompt

```
c:\>git config --global user.name "Akenarin Komkoon"
c:\>git config --global user.email codingthailand@gmail.com
```

Note: ในบางกรณีอาจมี library บางตัวต้องการ Python ก็ให้ติดตั้ง Python เวอร์ชัน 2 ได้ที่

<https://www.python.org/downloads/>

จริงๆ แล้วข้อ 4 สามารถติดตั้งที่หลังได้ เพราะเราสามารถทดสอบ App ผ่าน Browser เช่น Google Chrome ได้ หลังจากติดตั้ง Ionic เรียบร้อย ให้ตรวจสอบเวอร์ชันของ Ionic CLI (Command Line Interface) ด้วยคำสั่ง ionic -v

### █ Administrator: Command Prompt

```
C:\>ionic -v
3.20.0
```

หากต้องการดู Environments ต่างๆ ให้ cd เข้าไปในโฟลเดอร์โปรเจคก่อน จากนั้นใช้คำสั่ง ionic info และกด Enter ก็ได้เช่นเดียวกัน (ต้องสร้างโปรเจคใหม่ก่อน)

## การสร้างโปรเจคใหม่

การสร้างโปรเจคใหม่เราสามารถใช้ Ionic CLI พิมพ์คำสั่งได้เลย โดยมีรูปแบบคำสั่ง ดังนี้  
 ionic start <ชื่อโปรเจค> <template>

สำหรับการสร้างโปรเจคใหม่นั้นเราสามารถใส่ Templates หรือไม่ก็ได้ templates เป็นรูปแบบ App สำเร็จวูปทำให้ประหยัดเวลาในการพัฒนา ใน Ionic นั้นมี Templates ให้เลือก ดังนี้

- หากไม่ใส่ template จะเป็น App ในรูปแบบของ tabs application ใช้คำสั่ง

ionic start ionic3ebook

- Template แบบ blank ใช้คำสั่ง

ionic start ionic3ebook blank

- Template แบบ sidemenu ใช้คำสั่ง

ionic start ionic3ebook sidemenu

## ลงมือสร้างโปรเจคใหม่

ในหนังสือเล่มนี้เราจะใช้ Template sidemenu และตั้งชื่อโปรเจคว่า ionic3ebook ให้รันคำสั่งนี้ แล้วกด Enter (ชื่อโปรเจคสามารถตั้งชื่อตามความต้องการได้เลย)

```
ionic start ionic3ebook sidemenu
```

Note: สำหรับคนที่ใช้ Mac อย่าลืมใส่คำสั่ง sudo ไว้ด้านหน้าคำสั่งด้วย และเมื่อติดตั้งเสร็จแล้วให้ใช้คำสั่ง chmod -R 775 ไฟล์เดอร์โปรเจคด้วยนะ (เรื่องสิทธิ์เกี่ยวกับไฟล์และไฟล์เดอร์ของ MAC)

หลังจากสร้างโปรเจคเรียบร้อย ให้ใช้คำสั่ง cd ionic3ebook แล้วกด Enter เพื่อเข้าไปยังไฟล์เดอร์โปรเจคของเรา

```
C:\ Administrator: Node.js command prompt
```

```
C:\>cd ionic3ebook
```

```
C:\ionic3ebook>
```

Note: หลังจากติดตั้งจะมีการถามว่าเราต้องการ integrate กับ Cordova หรือไม่ ก็ให้ตอบ y ได้เลย ตั้งรูป

```
C:\>ionic start ionic3ebook sidemenu
Creating directory .\ionic3ebook - done!
Downloading and extracting sidemenu starter - done!

? Would you like to integrate your new app with Cordova to target native iOS and Android? (y/N) y-
```

Note: ตอนมาจะมีการถามว่าต้องการ connect กับ ionic dashboard (บริการ ionic cloud) หรือไม่ สามารถตอบ y แล้วกด enter เพื่อสมัคร และ link app ไปที่ cloud ได้เลย หรือจะตอบ n ก็ได้เช่นเดียวกัน เพราะสามารถสมัครทีหลังได้ผ่านเว็บนี้

<https://dashboard.ionicjs.com/signup>

เราสามารถใช้บริการ ionic dashboard นี้สำหรับใช้งาน ionic view (ໄว์เ薛ร์ app ให้ผู้ใช้ทดสอบ) หรือ build app ได้พร้อมๆ กัน

```
Supercharge your Ionic development with the Ionic Pro SDK
- Track runtime errors in real-time, back to your original TypeScript
- Push remote updates and skip the app store queue

Learn more about Ionic Pro: https://ionicframework.com/products

? Install the free Ionic Pro SDK and connect your app? (Y/n)
```

**Note:** เนื่องจากบางครั้ง Ionic 3 อาจมีอัปเดตเวอร์ชันย่อใหม่ อาจมีโค้ดเปลี่ยนบ้างในบางจุด นอกจากรคำสั่ง ionic start แล้ว สามารถดาวน์โหลด template เป็นได้ตามลิงก์นี้ <https://codingthailand.com/ionic3sidemenu.zip>

จะได้เวอร์ชันตรงกับหนังสือ เมื่อดาวน์โหลดเสร็จแล้ว ให้แตก zip ไฟล์ และ cd เข้าไปในโฟลเดอร์โปรเจค และวิ่งรันคำสั่ง npm install เพื่อติดตั้งแทนคำสั่ง ionic start ได้เช่นเดียวกัน

## เพิ่ม Platforms ที่เราต้องการ Build

ต่อมาเราควรเพิ่ม Platforms สำหรับ build app ถ้า Android ใช้คำสั่ง

```
ionic cordova platform add android
```

ถ้า iOS ใช้คำสั่ง (ต้องมีเครื่อง MAC)

```
ionic cordova platform add ios
```

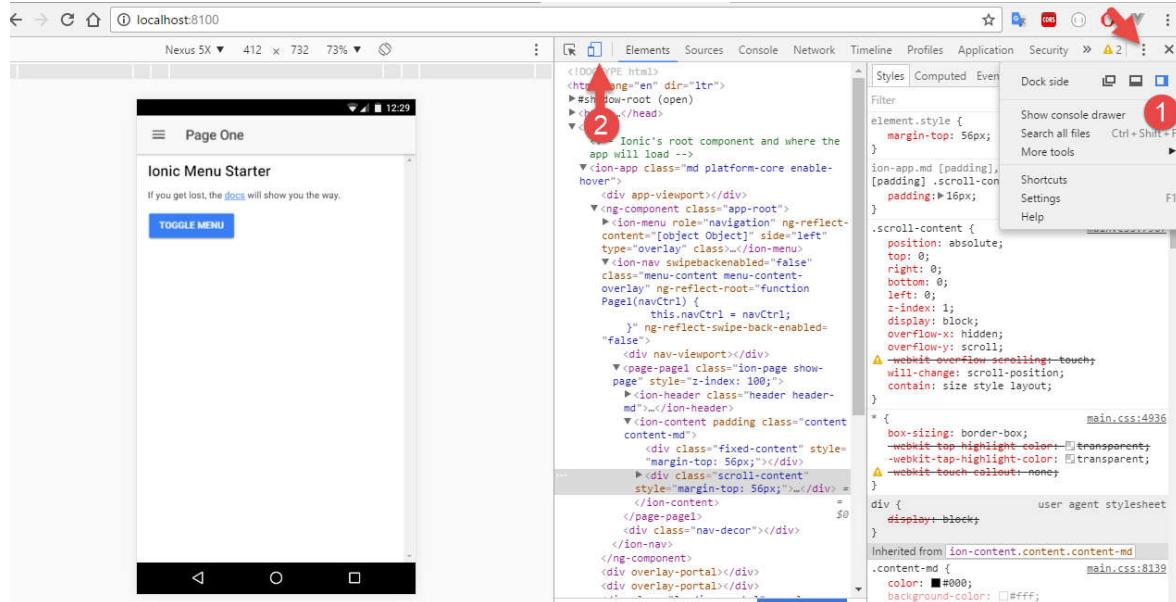
**Note:** การเพิ่ม platform ถ้าเป็น android เราต้องติดตั้ง Android Studio และต้องติดตั้ง Android SDK ก่อน

**Note:** หากต้องการยกเลิกหรือลบ platform ให้ใช้คำสั่ง ionic cordova platform rm android หรือ ionic cordova platform rm ios

**Note:** สำหรับคนที่ใช้ MAC ก่อนที่จะเพิ่ม platform ให้ติดตั้ง iOS simulator ก่อนด้วยคำสั่ง npm install -g ios-sim

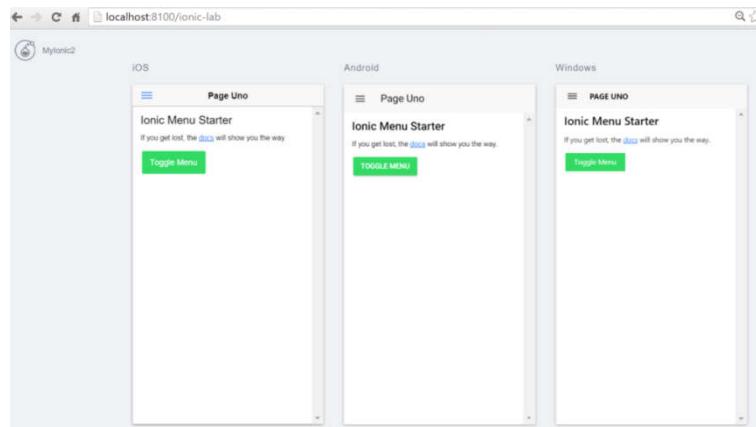
## การสั่งรัน Application ไปที่ Browser

การสั่งรัน Application ให้ใช้คำสั่ง ionic serve แล้วกด Enter หลังจากนั้นคำสั่งจะมีการเปิด Web browser ขึ้นมา หากเราแก้ไขโค้ดแล้วบันทึกไฟล์ Web browser จะ refresh อัตโนมัติ ในหนังสือเล่มนี้เราจะใช้ Google Chrome เป็นหลัก



หากต้องการยกเลิกคำสั่งรัน Application ให้กดปุ่ม Ctrl + C

อีกคำสั่งในการรัน Application ที่ไป Browser คือคำสั่ง ionic serve -l เป็นคำสั่งสำหรับสั่งรัน ionic lab เจ้าสามารถดู App พัฒนากันทั้งสาม platform ได้เลยทั้ง Android, iOS และ Windows



Note: เมื่อรันคำสั่ง ionic serve เสร็จแล้ว ให้ค้างหน้าจอ Command Prompt ไว้ตลอดเวลา หากเราแก้หรือเขียนโค้ดใหม่และบันทึกไฟล์ จะมีการ reload browser อัตโนมัติสามารถดูผลลัพธ์ได้เลย แต่หากมีการเขียนโค้ดผิด หรือมี errors จำนวนมาก บางครั้งเราอาจต้องใช้คำสั่ง ionic serve อีกรอบ

## การสั่งรัน Application ไปที่ เครื่องทดสอบจริง (ต่อ USB)

สำหรับ Android ใช้คำสั่ง ionic cordova run android --device

สำหรับ iOS ใช้คำสั่ง ionic cordova run ios --device

Note: การรันไปเครื่องทดสอบจริง อย่าลืมต่อสาย USB และเปิด Developer Options ในเครื่องด้วย ถ้าเป็น Android ปกติเข้าที่

เมนู Setting->About phone->Build Number (กดต่อเนื่อง 7 ครั้ง)->คลิกเครื่องหมายถูกที่ Stay awake->คลิกเครื่องหมายถูกที่

USB Debugging

## การสั่งรัน Application ไปที่ Emulator

การรัน App ไปที่ Emulator จะต้องเปิด Emulator ค้างไว้ก่อน จากนั้นค่อยรันคำสั่ง ดังนี้

สำหรับ Android ใช้คำสั่ง ionic cordova emulate android

สำหรับ iOS ใช้คำสั่ง ionic cordova emulate ios

## การอัปเดต Ionic CLI ให้เป็นเวอร์ชันล่าสุด

ให้ใช้ 3 คำสั่ง

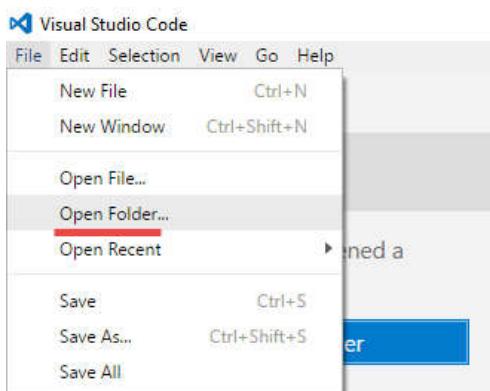
npm uninstall -g ionic@3

npm cache verify

npm install -g ionic@3

## การเปิดโปรเจค และเพิ่มเข้าใน Visual Studio Code

เปิด Visual Studio Code คลิกที่เมนู File -> Open Folder... แล้วเลือก โฟลเดอร์โปรเจคของเราได้เลย



```
14 "dependencies": {  
15   "@angular/common": "5.0.3",  
16   "@angular/compiler": "5.0.3",  
17   "@angular/compiler-cli": "5.0.3",  
18   "@angular/core": "5.0.3",  
19   "@angular/forms": "5.0.3",  
20   "@angular/http": "5.0.3",  
21   "@angular/platform-browser": "5.0.3",  
22   "@angular/platform-browser-dynamic": "5.0.3",  
23   "@ionic-native/core": "4.4.0",  
24   "@ionic-native/splash-screen": "4.4.0",  
25   "@ionic-native/status-bar": "4.4.0",  
26   "@ionic/pro": "1.0.20",  
27   "@ionic/storage": "2.1.3",  
28   "cordova-android": "6.3.0",  
29   "cordova-plugin-device": "^2.0.1",  
30   "cordova-plugin-ionic-keyboard": "^2.0.5",  
31   "cordova-plugin-ionic-webview": "^1.1.16",  
32   "cordova-plugin-splashscreen": "^5.0.2",  
33   "cordova-plugin-whitelist": "^1.3.3",  
34   "ionic-angular": "3.9.2",  
35   "ionicons": "3.0.0",  
36   "rxjs": "5.5.2",  
37   "sw-toolbox": "3.6.0",  
38   "zone.js": "0.8.18"  
--
```

หากมาถึงขั้นตอนนี้แสดงว่าเราพร้อมพัฒนา App เรียบร้อย!

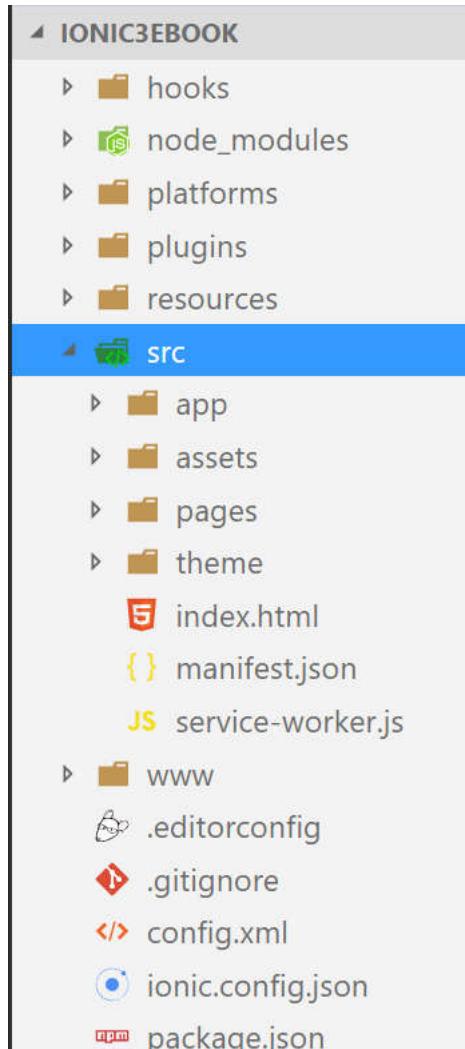
## บทที่ 4 แนะนำ Ionic Framework 3

ก่อนจะเข้าสู่การพัฒนา Mobile App ขอเกริ่นนำ Ionic เล็กน้อย และมาทำความรู้จักกับโครงสร้างไฟล์เดอร์ในโปรเจคกันก่อน

อย่างที่เกริ่นไปแล้ว Ionic เป็นรูปแบบการพัฒนา Mobile App ที่เรียกว่า Hybrid App Development คือ นักพัฒนาสามารถใช้ความรู้ด้าน Web Application ได้แก่ HTML5, CSS (Sass), JavaScript (JS ES6, Angular) มาพัฒนาเป็น Mobile App ได้

### โครงสร้างไฟล์ และไฟล์เดอร์ต่าง ๆ ในโปรเจค

หลังจากติดตั้ง Ionic เรียบร้อยมาดูกันว่า ไฟล์เดอร์สำคัญในโปรเจคของเรามีอะไรบ้าง



1. ไฟล์เดอร์ src เป็นไฟล์เดอร์ที่เราใช้งานบ่อยที่สุด ภายในไฟล์เดอร์ src ประกอบด้วย

- ไฟล์เดอร์ pages มีหน้าที่เก็บไฟล์เดอร์หน้าเพจทั้งหมดใน app ของเรา เรียกว่า page component ในแต่ละเพจจะประกอบด้วยไฟล์นามสกุล .html, .scss (ไว้ตกแต่งเพจ) และ .ts (ไว้เขียน logic ต่างๆ) หน้าเพจนี้สามารถ gen code ได้ด้วยการใช้ Ionic CLI
  - ไฟล์เดอร์ theme จะเก็บไฟล์นามสกุล .scss (Sass) มีไว้สำหรับแก้ไข และตกแต่งสีต่อ (Theme) ทั้งหมดใน App
  - ไฟล์เดอร์ app เป็นที่เก็บของ root component (app.component.ts) และเป็นจุดเริ่มการทำงานของ application
  - ไฟล์เดอร์ assets ใช้สำหรับเก็บรูปภาพ ไอคอน และไฟล์ JSON สำหรับใช้ใน App
2. ไฟล์เดอร์ www เป็นไฟล์เดอร์ในรูปแบบของ Web Application ซึ่งได้ผ่านการ build มาแล้ว เราจะต้องไม่แก้ไขไฟล์ต่างๆ ในไฟล์เดอร์นี้แต่ก็สามารถแก้ไขไฟล์ index.html ได้ (หากจำเป็น) ไฟล์เดอร์นี้เราสามารถ upload ขึ้น Server จริง และนำไปใช้ในรูปแบบ Web Application ได้ ไม่ใช่แค่ทำ Mobile App เท่านั้น เราเขียนครั้งเดียวจะได้ Web App มาใช้งานอีกด้วย
3. ไฟล์ config.xml เป็นไฟล์ตั้งค่าต่างๆ เพื่อ zobacายละเอียดให้ Cordova เพื่อ build app เราไปยัง Platforms ต่างๆ เช่น ชื่อของ Mobile App, เกอร์ชันของ Mobile App เป็นต้น
4. ไฟล์ package.json เป็นไฟล์ตั้งค่าของ npm ใช้สำหรับจัดการ และอัปเดตเวอร์ชัน Ionic และ dependencies ต่างๆ
5. ไฟล์เดอร์ resources มีไว้เก็บ splash screens และ icons ต่างๆ ใน App ของเราเวลาจะ build app การสร้าง resources นั้น สามารถใช้ Ionic CLI ในการสร้างไฟล์ icons และ splash screens อัตโนมัติได้
6. ไฟล์เดอร์ hooks เป็นไฟล์เดอร์ที่เก็บไฟล์เกี่ยวกับกระบวนการ build ต่างๆ โดยเราสามารถเพิ่ม custom script ในกระบวนการ build นี้ได้ สำหรับมือใหม่ยังไม่แน่นำให้แก้ไขในส่วนนี้
7. ไฟล์เดอร์ node\_modules เป็นไฟล์เดอร์ไว้เก็บ libraries ต่างๆ ที่ต้องใช้ในโปรเจกของเราร่วมถึง Angular กับ Ionic ด้วย
8. ไฟล์เดอร์ platforms เป็นไฟล์เดอร์เก็บไฟล์และโปรเจกของ mobile platforms ต่างๆ เมื่อเราเพิ่ม platform เข้ามา
9. ไฟล์เดอร์ plugins เป็นไฟล์เดอร์สำหรับเก็บ third party plugins ต่างๆ (native) ของ Cordova

## การใช้ IONIC CLI COMMANDS เพื่อสร้างโค้ดอัตโนมัติ

โดยส่วนใหญ่แล้วเราจะไม่ค่อยได้สร้างไฟล์ต่างๆ ของเพราเวมี Ionic CLI ช่วยในการ gen โค้ดต่างๆ ให้ (สะดวกมาก) ในหัวข้อนี้จะมาเรียนรู้วิธีการใช้ Ionic CLI กัน รูปแบบคำสั่ง คือ

ionic <command> <option>

รูปแบบคำสั่งจะขึ้นต้นด้วย ionic ตามด้วยคำสั่งที่เราต้องการ หากมี option ของคำสั่งนั้นก็สามารถใส่เข้าไปได้ เช่น ionic serve -l และมีหลายคำสั่งที่เราใช้กันไปบ้างแล้ว แต่ที่ใช้บ่อยมาก จะเป็นคำสั่งที่เรียกว่า Generate command เราสามารถ gen code ต่างๆ แบบอัตโนมัติโดยไม่ต้องพิมพ์เองทั้งหมด ได้แก่ การสร้าง page, component, directive, pipe, provider และ tabs รูปแบบคำสั่งมีดังนี้

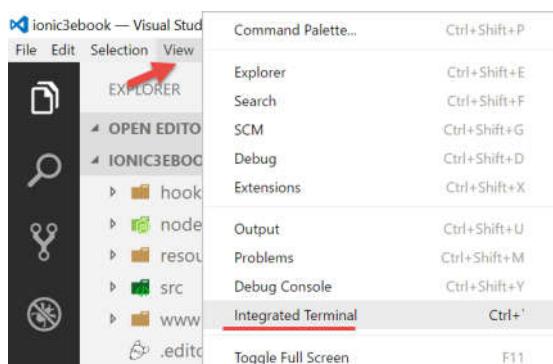
ionic g [page|component|directive|pipe|provider|tabs]

ยกตัวอย่างเช่น หากต้องการสร้างหน้าเพจใหม่ก็ให้ใช้คำสั่ง ionic g page product

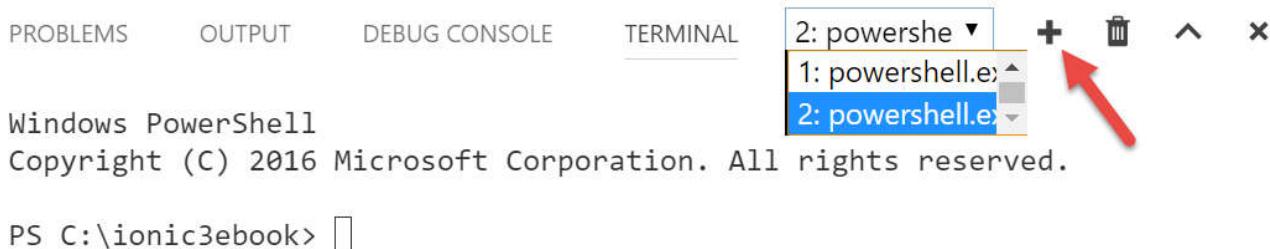
Note: หากต้องการดูคำสั่งเกี่ยวกับ generate ทั้งหมดให้พิมพ์ ionic g --list และเลื่อนปุ่มลูกศรที่คีย์บอร์ดเพื่อเลือกสิ่งที่ต้องการ generate ได้

## การใช้ Command line ภายในโปรแกรม Visual Studio Code

การพิมพ์คำสั่งเกี่ยวกับ Command line ทุกอย่าง ไม่ว่าจะเป็น ionic -v, ionic info, ionic serve หรือคำสั่งอื่นๆ สามารถพิมพ์คำสั่งเหล่านี้ได้ภายในโปรแกรม Visual Studio Code ได้โดย ให้คลิกที่เมนู **View -> Integrated Terminal**



และสามารถคลิกที่ปุ่ม (+) เพิ่มเพิ่มได้หลาย terminal อีกด้วย (ตัวนี้อาจເອົາໄວ້ວິນคำสั่ง ionic serve ຂອງເດືອກໄດ້)



## บทที่ 5 การสร้าง Pages, Basic Navigation และ Passing Data

### การสร้าง Pages

การสร้างหน้าเพจขึ้นมาใหม่ เราจะใช้ Ionic CLI ด้วยคำสั่ง `ionic g page <ชื่อเพจ>`

#### ทดลองสร้างเพจชื่อว่า About

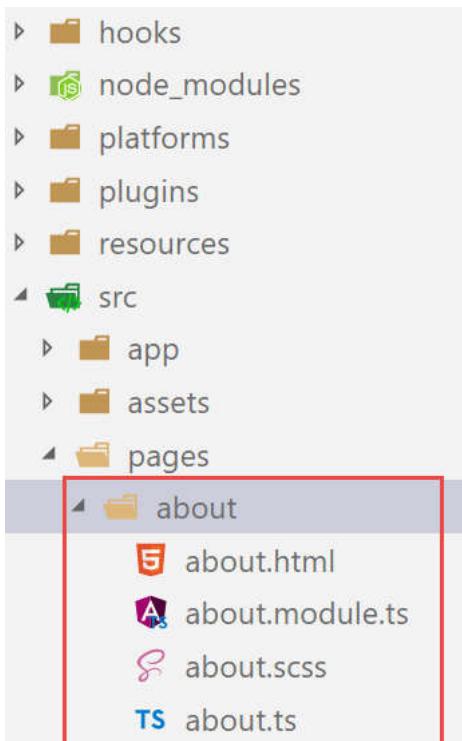
การสร้างเพจใหม่ มีขั้นตอน ดังนี้

- พิมพ์คำสั่ง `ionic g page about` และกด Enter

Administrator: Node.js command prompt

```
C:\ionic3ebook>ionic g page about
[OK] Generated a page named about!
```

- เสร็จแล้ว Ionic CLI จะสร้างโค้ดและสร้างไฟล์ให้เราอัตโนมัติ



- เปิดไฟล์ `src\app\app.module.ts` เพิ่มโค้ด `import { AboutPageModule } from '../pages/about/about.module';` เข้าไปด้านบน และกำหนดชื่อคลาสโมดูลของเพจเข้าไปในส่วนของ imports ดังรูป

```

10 import { SplashScreen } from '@ionic-native/splash-screen';
11
12 import { AboutPageModule } from '../pages/about/about.module'; ②
13
14 @NgModule({
15   declarations: [
16     MyApp,
17     HomePage,
18     ListPage
19   ],
20   imports: [
21     BrowserModule,
22     IonicModule.forRoot(MyApp),
23     AboutPageModule ③
24   ],
25   bootstrap: [IonicApp],
26   entryComponents: [
27     MyApp,
28     HomePage,
29     ListPage

```

4. เลือกเรียบร้อยสำหรับการสร้างเพจ อย่าลืมบันทึกไฟล์ด้วย

Note: ต้องไปหากต้องการ สร้าง Page ก็ให้ทำการขั้นตอนด้านบนนี้ได้เลย เป็นขั้นตอนที่เราต้องทำเป็นประจำ

#### ทดลองสร้างเมนู “เกี่ยวกับเรา” เพื่อเปิดหน้าเพจ About

หลังจากทดลองสร้างเพจ About เรียบร้อย มาลองสร้างเมนู (sidebar) เพื่อเปิดหน้าเพจ About กัน

1. เปิดไฟล์ src\app\app.component.ts เพิ่มโค้ด import { AboutPage } from './pages/about/about'; เข้าไปปิด้านบน جانนั้นเพิ่ม

Object อีก 1 ชุด โดยให้แก้ไข title และเพิ่มหน้า AboutPage เข้าไปในส่วนของ component ดังรูป

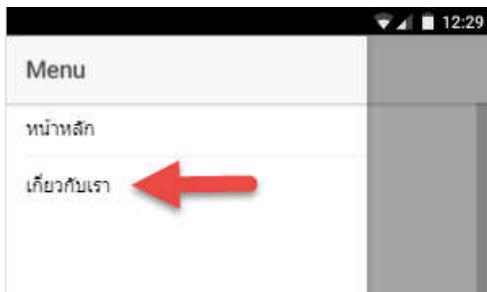
```

5
6 import { HomePage } from './pages/home/home';
7 //import { ListPage } from './pages/list/list';
8 import { AboutPage } from './pages/about/about'; ①
9
10 @Component({
11   templateUrl: 'app.html'
12 })
13 export class MyApp {
14   @ViewChild(Nav) nav: Nav;
15
16   //page: any = HomePage; ②
17   pages: Array<{title: string, component: any}>;
18
19
20 constructor(public platform: Platform, public statusBar: StatusBar)
21   this.initializeApp();
22
23
24 // used for an example of ngFor and navigation
25 this.pages = [
26   { title: 'หน้าหลัก', component: HomePage },
27   { title: 'เกี่ยวกับเรา', component: AboutPage } ②
28 ];

```

Note: ในหนังสือเล่มนี้เราจะไม่ได้ ListPage สามารถ comment ได้ด้วยวิธีคลิกไฟล์เดอร์ออกไปได้เลย

2. ทดสอบดูผลลัพธ์ที่ Browser (กรณีที่ยังไม่ได้รันคำสั่ง ionic serve ต้องรันคำสั่งนี้อีกครั้งหนึ่ง)



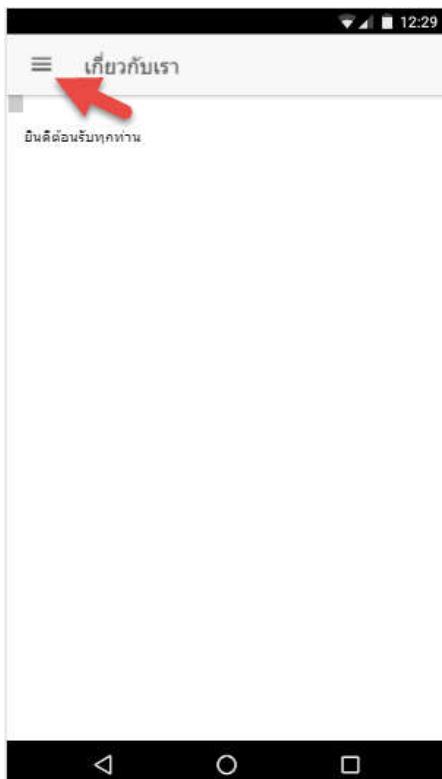
Note: การรันคำสั่ง ionic serve แนะนำให้รันทิ้งไว้เลยโดยให้เปิด Terminal อีก 1 ตัว หากมีการเขียนโค้ดแล้วเกิด error บางครั้งต้องสั่ง ionic serve ใหม่อีกครั้ง (ปกติไม่ต้อง serve ใหม่ทุกครั้ง)

3. หากต้องการให้หน้า About มีไอคอนเมนูด้วย ให้แก้ไขไฟล์และเพิ่มโค้ด about.html ดังนี้

```
<ion-header>
  <ion-navbar>
    <button ion-button menuToggle>
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-title>เกี่ยวกับเรา</ion-title>
  </ion-navbar>
</ion-header>

<ion-content padding>
  <p>ยินดีต้อนรับทุกท่าน</p>
</ion-content>
```

บันทึกไฟล์ และดูผลลัพธ์อีกครั้ง



Note: ຕ້ອໄປທາກມີການສ້າງເພິ່ນໃໝ່ ແລະ ອາຍາກເພີ່ມເມນູໃຫ້ລັບມາທບທວນໃນໜັກຂ້ອນນີ້ ທີ່ຈະ copy ສຳເນົາໄປໃຫ້ເລຸຍກີ່ຕໍ່

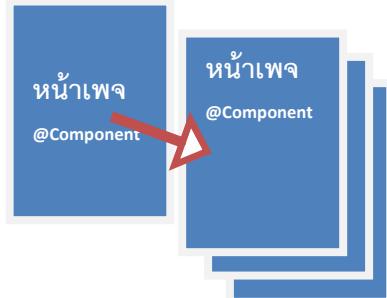
## ພິ່ນງຽນ Navigation

ເມື່ອເຮົາສ້າງເພິ່ນໄດ້ແລ້ວ ລຳດັບຕ່ອໄປຈະຕ້ອງທ່ານເຂົ້າໃຈເກືອງກັນ Navigation ລັດກາຮ່ອງ Navigation ຈະອູ່ໃນຢູ່ແບບ Stack ຄື່ອ ຊ້າມື່  
ການນຳເພິ່ນເຂົ້າມາແສດງ ຈະເຮືອກວ່າ push ແລະ ຊ້າມື່ການນຳເພິ່ນອອກໄປ ຈະເຮືອກວ່າ pop

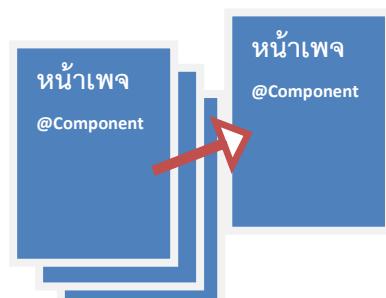
Navigation Stack



Push



Pop



ການທີ່ເຮົາຈະທຳ Navigation ຖ້າຈະຕ້ອງ import NavController ເຂົ້າມາກ່ອນ ແລະ ຕ້ອງປະກາດໄວ້ໃນສ່ວນຂອງ constructor ໃນຄລາສດ້ວຍ ໂດຍ  
ປັກຕິທາກເຮົາສ້າງເພິ່ນດ້ວຍ Ionic CLI (ຄໍາສັ່ງ ionic g page) ຈະມີການເພີ່ມໂຄດສ່ວນນີ້ເທົ່ອງໆແລ້ວ

```

TS about.ts ×

1 import { Component } from '@angular/core';
2 import { IonicPage, NavController, NavParams } from 'ionic-angular';
3
4 @IonicPage()
5 @Component({
6   selector: 'page-about',
7   templateUrl: 'about.html',
8 })
9 export class AboutPage {
10
11   constructor(public navCtrl: NavController, public navParams: NavParams) {
12 }
13
14   ionViewDidLoad() {
15     console.log('ionViewDidLoad AboutPage');
16   }
17
18 }

```

## การ Push หน้าเพจขึ้นมาแสดง

เพื่อการยกตัวอย่างให้เห็นภาพมากขึ้น เราจะสร้างปุ่มที่เพจ about เมื่อคลิกปุ่มแล้วให้เปิด (push) หน้าเพจ contact ขึ้นตอน มีดังนี้

- สร้างเพจ contact ก่อนโดยใช้คำสั่ง ionic g page contact และกด enter

ionic g page contact

- เปิดไฟล์ src\app\app.module.ts เพิ่มโค้ด `import { ContactPageModule } from '../pages/contact/contact.module';` เข้าไปด้านบน และกำหนดชื่อคลาสโมดูลของเพจเข้าไปในส่วนของ imports ดังรูป

app.module.ts

```

11
12 import { AboutPageModule } from '../pages/about/about.module';
13 import { ContactPageModule } from '../pages/contact/contact.module';
14
15 @NgModule({
16   declarations: [
17     MyApp,
18     HomePage,
19     ListPage
20   ],
21   imports: [
22     BrowserModule,
23     IonicModule.forRoot(MyApp),
24     AboutPageModule, ContactPageModule ②
25   ],

```

3. เปิดไฟล์ src\pages\contact\contact.html และติดเพิ่มไอคอน และบุํม menuToggle ดังนี้

```

<ion-header>
  <ion-navbar>
    <button ion-button menuToggle>
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-title>ติดต่อเรา</ion-title>
  </ion-navbar>
</ion-header>

<ion-content padding>
  <p>codingthailand.com</p>
</ion-content>

```

4. เปิดไฟล์ src\pages\about\about.html เพื่อเพิ่มบุํม, ไอคอน และเหตุการณ์ (event) คลิก พร้อมตั้งชื่อฟังก์ชันชื่อว่า goToContact()

```

<ion-header>
  <ion-navbar>
    <button ion-button menuToggle>
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-title>เกี่ยวกับเรา</ion-title>
  </ion-navbar>
</ion-header>

<ion-content padding>

```

```

<p>ยินดีต้อนรับทุกท่าน</p>
<br>
<button ion-button (click)="goToContact()" icon-left>
  <ion-icon name="person"></ion-icon> เปิดหน้าเพจติดต่อเรา
</button>

</ion-content>

```

5. เปิดไฟล์ src\pages\about\about.ts เพื่อ import เพจ contact เข้ามา แล้วเขียน method เพื่อเปิด (push) เพจ contact ดังนี้

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';

import { ContactPage } from '../contact/contact';

@IonicPage()
@Component({
  selector: 'page-about',
  templateUrl: 'about.html',
})
export class AboutPage {

  constructor(public navCtrl: NavController, public navParams: NavParams) {
  }

  ionViewDidLoad() {
    console.log('ionViewDidLoad AboutPage');
  }

  goToContact() {
    this.navCtrl.push(ContactPage);
  }
}

```

อธิบายโดย ส่วนของด้านบนเราได้ Import คลาส ContactPage เข้ามา จากนั้นสร้าง method ชื่อว่า goToContact() เมื่อผู้ใช้คลิกปุ่ม ก็จะมาทำงานใน method หรือพังก์ชัน นี้ เพื่อเปิดเพจ Contact ด้วยคำสั่ง this.navCtrl.push(ContactPage);

```

about.html
1 <ion-header>
2   <ion-navbar>
3     <button ion-button menuToggle>
4       <ion-icon name="menu"></ion-icon>
5     </button>
6     <ion-title>เกี่ยวกับเรา</ion-title>
7   </ion-navbar>
8 </ion-header>
9
10 <ion-content padding>
11   <p>ยินดีต้อนรับทุกท่าน</p>
12   <br>
13   <button ion-button (click)="goToContact()">
14     <ion-icon name="person"></ion-icon>
15     เปิดหน้าเพจติดต่อเรา
16   </button>
17 </ion-content>
18

```

```

about.ts
1 import { Component } from '@angular/core';
2 import { NavController, NavParams } from 'ionic-angular';
3
4 import { ContactPage } from '../contact/contact';
5
6 @Component({
7   selector: 'page-about',
8   templateUrl: 'about.html'
9 })
10 export class AboutPage {
11
12   constructor(public navCtrl: NavController, public
13
14   goToContact() {
15     this.navCtrl.push(ContactPage);
16   }
17 }
18

```

Note: หากต้องการ push หน้าไหนก็ให้ import และใช้ชื่อคลาสของเพจให้กับ argument ของ method push() นั้นเอง

## การ Pop หน้าเพจ

หลังจากที่เราได้เปิดหน้าเพจ Contact แล้วมาลอง pop ออกกัน จะง่ายแค่ถักในการ pop ก็ใช้งานคล้ายกันกับ push ดังนี้

```
this.nav.pop();
```

ตัวอย่าง

```

class ContactPage {

constructor(public navCtrl: NavController) { }

goBack() {

  this.navCtrl.pop();

}

}

```

แต่มีอีกวิธีหากไม่ต้องการสร้างฟังก์ชัน หรือ method เพิ่ม เรายสามารถระบุ attribute หรือ directive ชื่อว่า navPop เข้าไปที่บูมได้เลย (ไม่ต้องเขียนโค้ดที่ไฟล์ ts) มาลองเขียนกันดูกัน

- เปิดไฟล์ src\pages\contact\contact.html และกดไดด์ ดังนี้

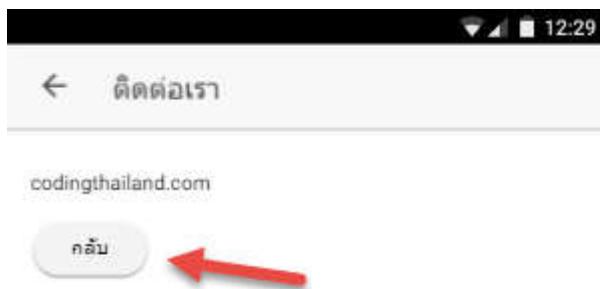
```

<ion-header>
  <ion-navbar>
    <button ion-button menuToggle>
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-title>ติดต่อเรา</ion-title>
  </ion-navbar>
</ion-header>

<ion-content padding>
  <p>codingthailand.com</p>
  <button ion-button color="light" round navPop>กลับ</button>
</ion-content>

```

- บันทึกไฟล์ และลองคลิกปุ่ม “เปิดหน้าเพจติดต่อเรา” ที่เพจ About จากนั้นคลิกปุ่ม “กลับ” แค่นี้เราก็ pop เพจได้เรียบร้อย



### การกำหนดเพจ ให้เป็น root page

จะสังเกตว่าเวลาที่เรา push หน้าเพจใหม่ตาม เพจนั้นจะมาทับอยู่บนเพจเดิมที่มีอยู่ และตัวมันเองสามารถ pop ได้ แต่ถ้าหากเราต้องการ เปิดเพจใดๆ แล้วอย่างให้เป็น root page (เพจหลัก) ให้ใช้คำสั่ง this.navCtrl.setRoot(ชื่อคลาสเพจ);

ให้ลองแก้ไขไฟล์ src\pages\about\about.ts ที่ method goToContact() และลองรันดูอีกครั้ง

```

goToContact() {
  //this.navCtrl.push(ContactPage);
  this.navCtrl.setRoot(ContactPage);
}

```

Note: เราไม่ควรใช้คำสั่ง setRoot() เพื่อกลับส่วนหนึ่งหรือหน้าหลักของ App

## การส่งข้อมูลระหว่างเพจ (Passing Data between Pages)

หัวข้อนี้เป็นหัวข้อที่สำคัญมาก เพราะเราจะได้ใช้ประโยชน์ในการพัฒนา Mobile App เช่น การทำเพจในรูปแบบของ Master Detail ถ้าเราไม่หน้าเพจสินค้า หากคลิกที่ชื่อสินค้า เราสามารถส่งรหัสสินค้าไปที่เพจรายละเอียดสินค้าเพื่อดึงข้อมูลจากฐานข้อมูลมาแสดง เป็นต้น

ในการส่งข้อมูลข้ามเพจนั้น ที่เพจหลัก (Master) เราสามารถส่งข้อมูลไปกับ method push() ได้เลย (หรือ method setRoot() ก็ได้) ในตัวอย่างนี้เราจะทดลองส่งข้อมูลจากเพจ About ไปยังเพจ Contact

1. เปิดไฟล์ src\pages\about\about.ts เราจะทดลองส่งข้อมูล 2 ตัว ได้แก่ companyName กับ companyWebsite โดยแก้ไข method goToContact() ดังนี้

```
goToContact() {
  this.navCtrl.push(ContactPage, {
    companyName: 'CodingThailand',
    companyWebsite: 'https://codingthailand.com'
  });
  //this.navCtrl.setRoot(ContactPage);
}
```

2. เปิดไฟล์ src\pages\contact\contact.ts ซึ่งเราจะใช้หน้านี้สำหรับรับข้อมูลจากเพจ About ในการรับข้อมูลที่ถูกส่งมา เราจะต้อง import NavParams ก่อน และต้องกำหนดที่ constructor ด้วย (ปกติถ้าใช้การ gen ได้จะมี NavParams มาให้โดย) ในการเรียกข้อมูลที่รับมานั้นให้ใช้ method ที่ชื่อว่า get() และต้องมีการประกาศตัวแปรในคลาสเพื่อรับค่าที่ส่งมาด้วย

```
import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';

@IonicPage()
@Component({
  selector: 'page-contact',
  templateUrl: 'contact.html',
})
export class ContactPage {
  companyName:string;
  companyWebsite:string;

  constructor(public navCtrl: NavController, public navParams: NavParams) {
    this.companyName = this.navParams.get('companyName');
    this.companyWebsite = this.navParams.get('companyWebsite');
  }
}
```

เปิดไฟล์ src\pages\contact\contact.html และเพิ่มโค้ดเพื่อนำตัวแปรในคลาส (ที่รับข้อมูลมา) ไปแสดงผลที่หน้าเพจ ดังนี้

```

<ion-header>
  <ion-navbar>
    <button ion-button menuToggle>
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-title>ติดต่อเรา</ion-title>
  </ion-navbar>
</ion-header>

<ion-content padding>
  <p>codingthailand.com
  <br>
  ชื่อบริษัท: {{companyName}} <br>
  เว็บไซต์: {{companyWebsite}}
  </p>
  <button ion-button color="light" round navPop>กลับ</button>
</ion-content>

```

Note: เครื่องหมายวงเล็บปีกกาคู่ {{companyName}} เป็นหลักการของภาษาผู้กู้ข้อมูลระหว่างตัวแปรในคลาส และ template ของ Angular เรียกว่า Data Binding (interpolation) อ่านเพิ่มเติมได้ที่ <https://angular.io/docs/ts/latest/guide/template-syntax.html>

```

contact.html
1  <ion-header>
2    <ion-navbar>
3      <button ion-button menuToggle>
4        <ion-icon name="menu"></ion-icon>
5      </button>
6      <ion-title>ติดต่อเรา</ion-title>
7    </ion-navbar>
8  </ion-header>
9
10 <ion-content padding>
11   <p>
12     codingthailand.com
13   <br>
14   ชื่อบริษัท: {{companyName}} <br>
15   เว็บไซต์: {{companyWebsite}}
16   </p>
17   <button ion-button color="light"
18     กลับ
19   </button>
20 </ion-content>

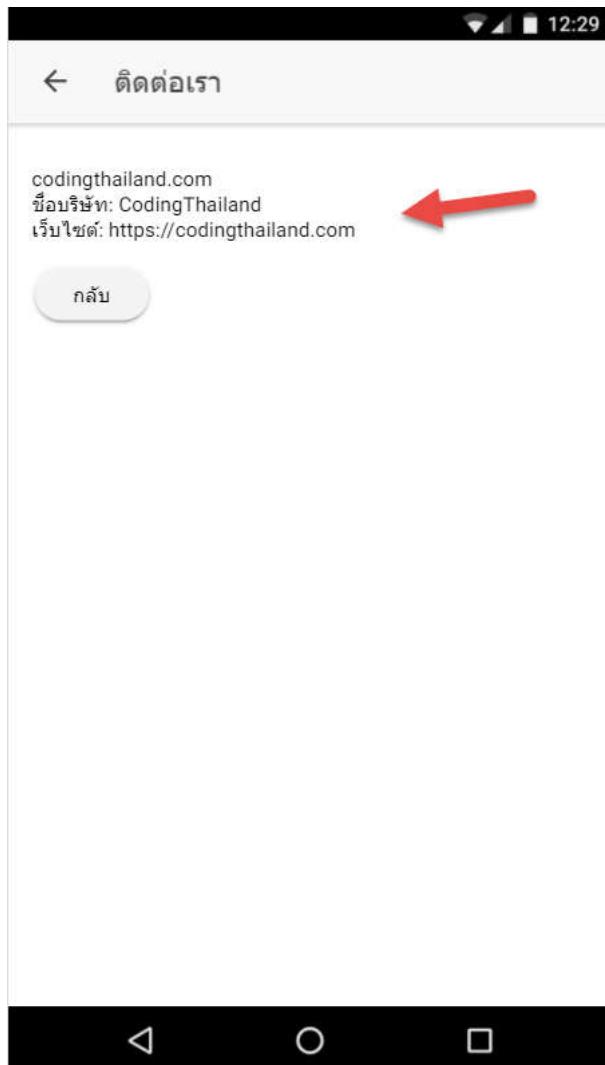
```

```

contact.ts
1 import { Component } from '@angular/core';
2 import { NavController, NavParams } from 'ionic-angular';
3
4 @Component({
5   selector: 'page-contact',
6   templateUrl: 'contact.html'
7 })
8 export class ContactPage {
9   companyName:string;
10  companyWebsite:string;
11
12  constructor(public navCtrl: NavController,
13    this.navParams.get('
14      this.companyName = this.navParams.get(
15    )
16  }
17

```

3. ทดสอบ App แล้วลองคลิกปุ่ม “เปิดหน้าเพจติดต่อเรา” ในเบน About ดู จะเห็นว่ามีการส่งข้อมูลข้ามมายังเพจ Contact เรียบร้อย



## บทที่ 6 Ionic Components

### Components คืออะไร

Components คือ ชุดของ User Interface (UI) สำหรับที่เราสามารถนำมาใช้ได้ทันที เช่น Buttons, Cards, Alerts, Modals, Lists เป็นต้น โดยใน Component แต่ละตัวจะมีข้อกำหนด พื้นฐาน และการใช้งานที่แตกต่างกัน สามารถดูรายละเอียด และทดลองดูตัวอย่าง Components ทั้งหมดได้ที่ <http://ionicframework.com/docs/components/>

### การใช้ Components เป็นต้น

เนื่องจาก Component มีหลายตัว ในหนึ่งสีก็เล่นนี้จะยกตัวอย่างเฉพาะที่ใช้บ่อย แต่ถ้าหัดลักษณะแล้วสามารถประยุกต์ใช้ได้กับทุกตัว

#### Buttons

เป็นปุ่มกด สามารถใช้งานได้ง่ายมาก เพียงแค่คำสั่ง <button ion-button>Button</button> หากต้องการปรับสีปุ่มต่างๆ ก็เขียนโค้ดได้ดังนี้

```
<button ion-button color="light" round>Light Round</button>
<button ion-button round>Primary Round</button>
<button ion-button color="secondary" round>Secondary Round</button>
<button ion-button color="danger" round>Danger Round</button>
<button ion-button color="dark" round>Dark Round</button>
```

ดูรายละเอียดการใช้งานทั้งหมดได้ที่ <https://ionicframework.com/docs/components/#buttons>

#### Icons

ใน Ionic จะมีไอคอนสำหรับต่างๆ มาให้เราใช้เรียบร้อย โดยมีการใช้งานดังนี้

```
<ion-icon name="heart"></ion-icon>
```

หากอยากรีเซ็ตไอคอนแค่รูปชื่อ (name) เข้าไปได้ตามต้องการ ดูไอคอนทั้งหมดได้ที่นี่

<https://ionicframework.com/docs/ionicons/>

## Cards

Card มีโครงสร้างแสดงผลข้อมูลในหน้าเพจโดยแบ่งเป็นส่วนๆ และสามารถ มีวิธีการใช้งานง่ายๆ ดังนี้

```
<ion-card>
  <ion-card-header>
    Card Header
  </ion-card-header>

  <ion-card-content>
    <!--เพิ่มเนื้อหาข้อมูลที่นี่! -->
  </ion-card-content>

</ion-card>
```

ดูรายละเอียดการใช้งาน Cards ทั้งหมดได้ที่ <http://ionicframework.com/docs/components/#cards>

## Grid

เราสามารถแบ่งหน้าเพจในรูปแบบของคอลัมน์ได้ โดยสามารถบุกร่องให้แต่ละคอลัมน์จะให้กว้างกี่เพอร์เซ็นต์ หากใครเคยใช้ CSS Framework เช่น Bootstrap จะมีแนวคิดการใช้แบบเดียวกัน รายละเอียดการใช้ดูได้ที่ <http://ionicframework.com/docs/components/#grid>

มาลองสร้างเมนูหน้าแรกด้วย Grid กัน

1. เปิดไฟล์ src\pages\home\home.html ให้แก้ไขโค้ดหน้าเพจ ดังนี้

```
<ion-header>
  <ion-navbar>
    <button ion-button menuToggle>
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-title>หน้าหลัก</ion-title>
  </ion-navbar>
</ion-header>

<ion-content padding>

  <ion-grid>
    <ion-row>
      <ion-col width-50>
```

```

        <button ion-button block large icon-left>
            <ion-icon name="home"></ion-icon>
            หน้าหลัก
        </button>
    </ion-col>
    <ion-col width-50>
        <button ion-button block large icon-left>
            <ion-icon name="aperture"></ion-icon>
            เกี่ยวกับ
        </button>
    </ion-col>
</ion-row>
<ion-row>
    <ion-col width-75>
        <button ion-button block large icon-left>
            <ion-icon name="person"></ion-icon>
            ติดต่อเรา
        </button>
    </ion-col>
    <ion-col width-25>
        <button ion-button block large icon-only>
            <ion-icon name="settings"></ion-icon>
        </button>
    </ion-col>
</ion-row>

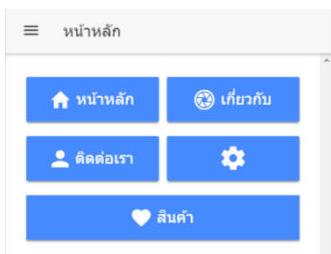
<ion-row>
    <ion-col>
        <button ion-button block full large icon-left>
            <ion-icon name="heart"></ion-icon>
            สินค้า
        </button>
    </ion-col>
</ion-row>

</ion-grid>

</ion-content>

```

## 2. บันทึกไฟล์ แล้วรันดูอีกครั้ง



Components อื่นๆ มักใช้กับฟอร์ม หรือเกี่ยวข้องกับ Data ต่างๆ จะมีตัวอย่างใช้งานแทรกในบทต่อไป

## บทที่ 7 การเรียกใช้ APIs ด้วย Angular HTTP Services

ในการพัฒนา Mobile App ปอยครั้งที่ต้องเขียนติดต่อกับฐานข้อมูล หรือติดต่อกับระบบ Backend ต่างๆ อาจมีการเรียกดูข้อมูลเพื่อนำมาแสดงผล การเพิ่มข้อมูล การแก้ไขข้อมูล หรือลบข้อมูล สำหรับ Ionic นั้นจะใช้หลักการของ Angular Services (HttpClient) เพื่อช่วยติดต่อและจัดการกับข้อมูลต่างๆ ในฝั่งของ Backend ให้กับเรา

Note: รายละเอียดเชิงลึกของ HTTP Services ได้ที่ <https://angular.io/guide/http>

### ขั้นตอนการใช้ HTTP Services หรือ provider ใน Ionic

การใช้ HTTP Services นั้น ใน Ionic จะเรียกอีกอย่างหนึ่ง provider เราสามารถใช้ Ionic CLI ช่วยสร้างได้ด้วยคำสั่ง `ionic g provider <ชื่อ provider>`

Note: การใช้ HTTP Services นอกจากจะจัดการข้อมูลฝั่ง Backend แล้ว ยังสามารถจัดการกับไฟล์ JSON ได้ด้วย

ส่วนใหญ่แล้วในฝั่ง Backend จะคืนค่าข้อมูลมาให้ App ของเราในรูปแบบ JSON มาทดลองเขียนกันเลย!

หนังสือเล่มนี้ได้เตรียมข้อมูลฝั่ง Backend ให้เป็นตัวอย่างแล้ว ซึ่งเขียนด้วยภาษา PHP แบบง่ายๆ โดยเราจะทดลองดึงข้อมูลผ่าน URL นี้ [https://codingthailand.com/api/get\\_courses.php](https://codingthailand.com/api/get_courses.php)

Note: การเขียน Backend สามารถใช้ภาษาคาวาเรกได้ตามที่เราถนัด ขอให้ return ข้อมูลที่ได้ออกมาเป็น JSON

ไฟล์ `get_courses.php` นี้จะทำดึงข้อมูลรายชื่อคอร์สเรียนพร้อมจากเว็บ codingthailand.com และคืนคามาในรูปแบบ JSON ตัวอย่างการเขียน php มีดังนี้

```
<?php
    header("Access-Control-Allow-Origin: *");
    $link = mysqli_connect('localhost', 'root', '', 'mydb');

    mysqli_set_charset($link, 'utf8');
    $sql = "SELECT * FROM courses";
```

```

$result = mysqli_query($link, $sql);

$coursesArray = array();

while ($row = mysqli_fetch_assoc($result)) {

    $coursesArray[] = $row;

}

echo json_encode($coursesArray);

?>

```

ทดสอบรัน [https://codingthailand.com/api/get\\_courses.php](https://codingthailand.com/api/get_courses.php) ที่ Browser จะได้ผลลัพธ์ดังนี้



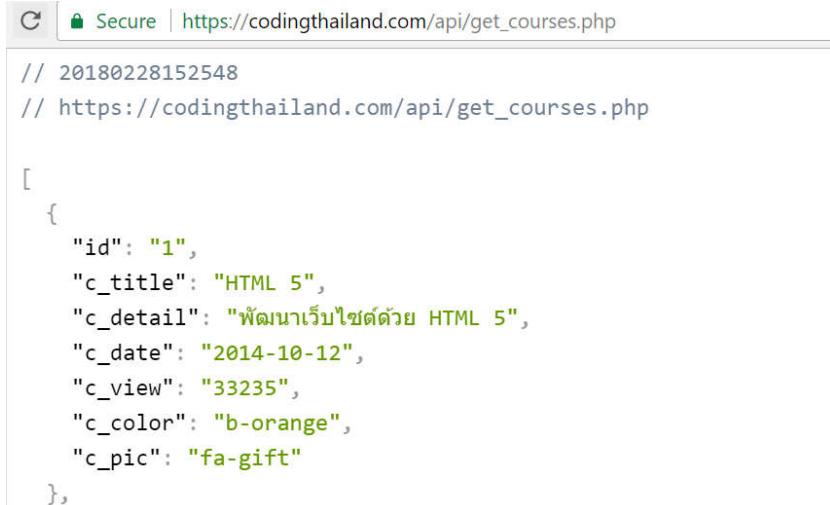
```
[{"id": "1", "c_title": "HTML 5", "c_detail": "\u0e1e\u0e31\u0e12\u0e19\u0e32\u0e40\u0e27\u0e47\u0e1a\u0e44\u0012", "c_view": "16249", "c_color": "b-orange", "c_pic": "fa-gift"}, {"id": "2", "c_title": "CSS 3", "c_detail": "\u0e1e\u0e31\u0e12\u0e19\u0e32\u0e41\u0e25\u0e30\u0e15\u0e01\u0e41\u0e15\u0e48\u0e07\u0e43", "c_date": "2014-10-13", "c_view": "9226", "c_color": "b-red", "c_pic": "fa-comment-o"}, {"id": "3", "c_title": "PHP Basic", "c_detail": "\u0e40\u0e02\u0e35\u0e22\u0e19\u0e42\u0e1b\u0e23\u0e41\u0e01\u0e23\u0e21\u0e14\u0e45\u0e1e\u0e37\u0e49\u0e19\u0e10\u0e32\u0e19", "c_date": "2014-10-12", "c_view": "13693", "c_color": "b-green", "c_OOP": "OOP", "c_detail": "\u0e40\u0e23\u0e35\u0e22\u0e19\u0e01\u0e32\u0e23\u0e40\u0e02\u0e35\u0e22\u0e19\u0e42 PHP", "c_date": "2014-10-12", "c_view": "5395", "c_color": "b-black", "c_pic": "fa-certificate"}, {"id": "5", "c_title": "MySQL", "c_detail": "\u0e2a\u0e23\u0e49\u0e32\u0e07\u0e40\u0e27\u0e47\u0e1a\u0e44\u0e0b\u0e15\u0e4c\u0e10\u0e32\u0e19\u0e02\u0e49\u0e2d\u0e21\u0e39\u0e25 MySQL", "c_date": "2014-10-12", "c_view": "17231"}]
```

จากภาพ จะเห็นว่า เราได้ข้อมูลจากฐานข้อมูลมา โดยแต่ละแถวจะมีคอลัมน์ และค่าข้อมูลแสดงออกมา ในส่วนของ Ionic เวลาข้างถึงคอลัมน์ในฐานข้อมูลก็อย่าลืมเขียนและอ้างชื่อให้ตรงกันด้วย

หากต้องการจัดรูปแบบ JSON ให้สวยงามสามารถ Google Chrome Extension ชื่อว่า JSON Viewer ได้ที่นี่

<https://github.com/tulios/json-viewer>

เมื่อทดสอบจริงๆเฟรชเพจอีกครั้งจะได้รูปแบบ JSON ที่สวยงาม ดังรูป

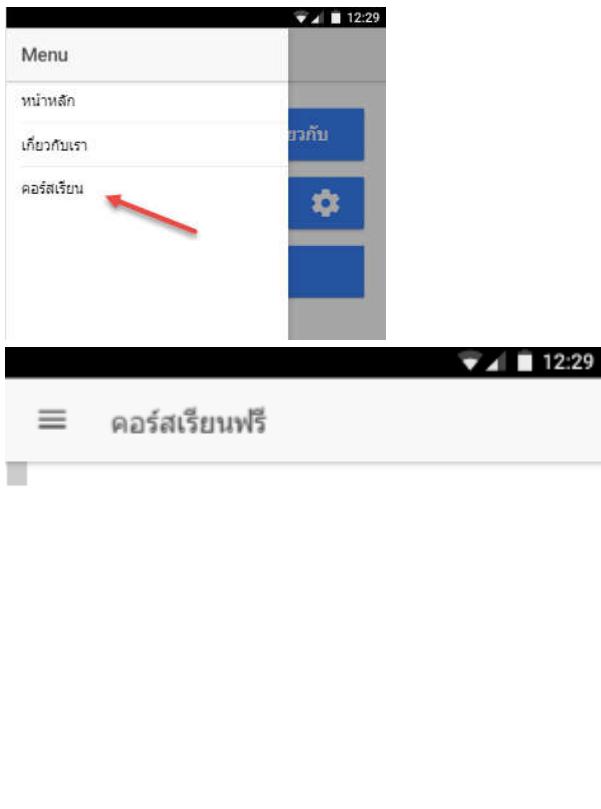


```
// 20180228152548
// https://codingthailand.com/api/get_courses.php

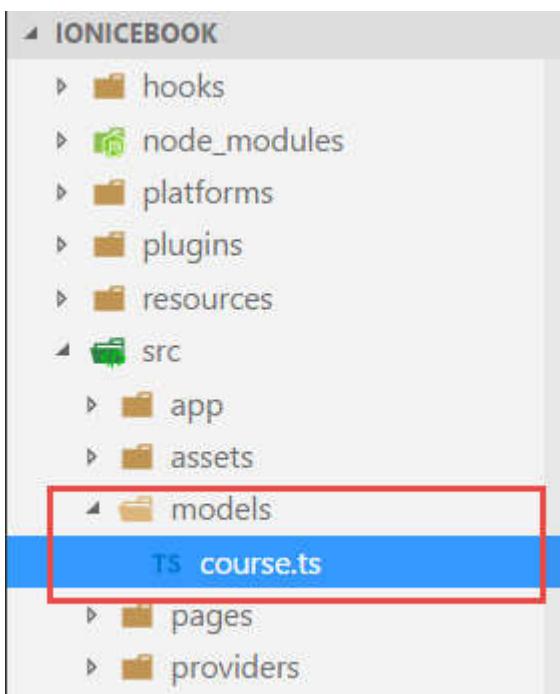
[
  {
    "id": "1",
    "c_title": "HTML 5",
    "c_detail": "พัฒนาเว็บไซต์ด้วย HTML 5",
    "c_date": "2014-10-12",
    "c_view": "33235",
    "c_color": "b-orange",
    "c_pic": "fa-gift"
  },
  ...
]
```

ในหัวข้อนี้เราจะดึงข้อมูลมาแสดงในรูปแบบ Lists กัน มีขั้นตอน ดังนี้

1. สร้างเพจใหม่ ด้วยคำสั่ง ionic g page course (ให้ทดลอง import CoursePageModule ในไฟล์ app.module.ts ด้วยตัวเอง)
2. สร้างเมนูด้านข้าง (ให้ทดลองเชียนเพิ่มเมนูด้านข้างด้วยตัวเอง ที่ไฟล์ app.component.ts) ดังรูป



3. สร้าง model เพื่อรองรับข้อมูลที่ได้จาก Backend โดยให้คลิกขวาที่ไฟล์เดอร์ src เลือก New Folder จากนั้นพิมพ์ชื่อไฟล์เดอร์ ใหม่ว่า models และสร้างไฟล์ course.ts ด้านใน ดังรูป



เปิดไฟล์ models/course.ts แล้วพิมพ์โค้ดเพื่อสร้าง model โดยตั้งชื่ออินเตอร์เฟสว่า Course และกำหนดตัวแปรต่างๆให้ตรงกับข้อมูล (JSON) ที่ได้รับมาจาก Backend ดังนี้

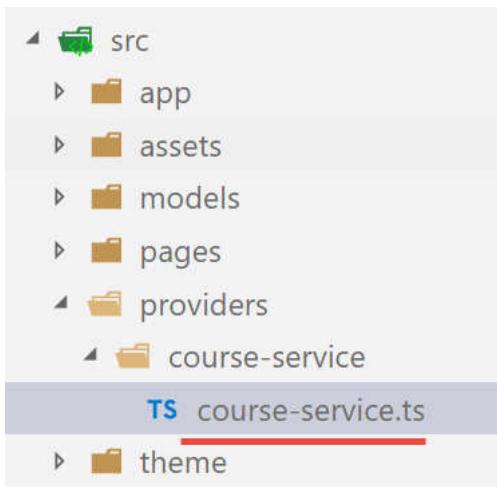
```
export interface Course {
  id: string;
  c_title: string;
  c_detail: string;
  c_date: string;
  c_view: string;
  c_color: string;
  c_pic: string;
}
```

```
1  export interface Course {
2    id: string;
3    c_title: string;
4    c_detail: string;
5    c_date: string;
6    c_view: string;
7    c_color: string;
8    c_pic: string;
9  }
10
```

4. สร้าง provider ใหม่ ด้วยคำสั่ง ionic g provider courseService แล้วกด Enter

**ionic g provider courseService**

Path ของ provider ที่สร้าง จะอยู่ที่ src\providers\ชื่อไฟล์.ts



5. เปิดไฟล์ src\providers\course-service\course-service.ts เขียนคำสั่ง ดังนี้

```

import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';

import { Course } from '../../models/course';
import { Observable } from 'rxjs/Observable';

@Injectable()
export class CourseServiceProvider {
  apiUrl:string = 'https://codingthailand.com/api/get_courses.php';

  constructor(public http: HttpClient) { }

  //เขียน method ชื่อว่า getCourse() เพื่อสำหรับดึงข้อมูลคอร์สเรียน
  //กำหนด this.http.get<Course[]>(this.apiUrl); ก็อกำสั่งให้ดึงข้อมูลจาก Backend ด้วย
  method get() ตาม URL ที่ได้ประกาศไว้ด้านบน
  //และทำการแปลง json จากที่ได้จาก backend ให้กับโมเดล Course
  //จากนั้นก็ return ข้อมูลคอร์สเรียนทั้งหมดออกไปกับโมเดล Course ในรูปแบบ Observable
  //ทำให้เพื่อต่างๆสามารถเรียกใช้เมธอด getCourse() ของ Provider นี้ได้
  getCourse(): Observable<Course[]> {
    return this.http.get<Course[]>(this.apiUrl);
  }
}

```

6. เมื่อสร้าง provider หรือ Service เรียบร้อยแล้ว ขั้นตอนต่อมาที่ต้องตรวจสอบ คือ เมื่อไหร่ก็ตามที่จะใช้งาน HTTP Service จะต้องมีการ import HttpClientModule เข้ามาใช้งานใน App ของเราด้วย (ทำแค่ครั้งเดียว) โดยให้เปิดไฟล์ src\app\app.module.ts เพื่อ import และ กำหนดในส่วนของ imports ดังรูป (service แล้วอย่าลืมบันทึกไฟล์) ซึ่งส่วนนี้จะต้อง CourseServiceProvider จะต้องถูก import เข้ามาในไฟล์ และกำหนดที่ส่วนของ providers ด้วย (ปกติส่วนนี้จะถูกกำหนดโดยอัตโนมัติ) สังเกตว่าจะมีโค้ดส่วนนี้อยู่แล้ว

```

14 import { CoursePageModule } from '../pages/course/course.module';
15 import { CourseServiceProvider } from '../providers/course-service/course-service';
16
17 import { HttpClientModule } from '@angular/common/http';
18
19 @NgModule({
20   declarations: [
21     MyApp,
22     HomePage,
23     ListPage
24   ],
25   imports: [
26     BrowserModule,
27     HttpClientModule, ← (2)
28     IonicModule.forRoot(MyApp),
29     AboutPageModule, ContactPageModule, CoursePageModule
30   ],
31   bootstrap: [IonicApp],
32   entryComponents: [
33     MyApp,
34     HomePage,
35     ListPage
36   ],
37   providers: [
38     StatusBar,
39     SplashScreen,
40     {provide: ErrorHandler, useClass: IonicErrorHandler},
41     CourseServiceProvider
42 ]

```

7. ต่อมาหากเราต้องการเรียกใช้งาน provider ที่สร้างขึ้น จะต้องมีการ import provider นั้นเข้ามาใช้งานในเพจ และต้องประกาศ (inject) ที่ constructor ของเพจนั้นๆ ด้วย จากนั้นต้อง import model เข้ามาด้วยเช่นกันเพื่อรอวับข้อมูลที่ return มาจาก Provider ในตัวอย่างนี้เราจะดึงข้อมูลคอร์สเรียนทั้งหมดจาก Backend มาแสดงผลในเพจ course ให้เปิดไฟล์ src\pages\course\course.ts เพื่อเรียกใช้ provider เข้ามาใช้งาน เพิ่มคำสั่ง ดังนี้

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';

import { CourseServiceProvider } from '../../providers/course-
service/course-service'; //import service เข้ามาใช้งาน

```

```

import { Course } from '../../../../../models/course'; //import model เข้ามาใช้งาน

import { Subscription } from 'rxjs/Subscription'; //import Subscription
เพื่อ unsubscribe() ข้อมูลจาก Server

@IonicPage()
@Component({
  selector: 'page-course',
  templateUrl: 'course.html'
})
export class CoursePage {
  courses: Course[]; //ประมวลผล courses มีชนิดข้อมูลเป็นโมเดล Course (array)
  sub: Subscription;

  constructor(public navCtrl: NavController,
              public navParams: NavParams,
              private courseServiceProvider: CourseServiceProvider
            ) {}

  //สร้าง method ที่อ่าน getCourse() มีหน้าที่ดึงค่ามาจาก Backend
  //subscribe ข้อมูลที่ได้รับจาก service (CourseServiceProvider)
  //และส่งค่าให้กับตัวแปร courses (courses (res) => this.courses = res)
  private getCourses() {
    this.sub = this.courseServiceProvider.getCourse().subscribe(
      (courses: Course[]) => this.courses = courses //courses ทางที่ส่งมาจากคอร์ส
      ที่มาจากการดึง Backend นั้นเอง
    );
  }

  //ionViewWillEnter() เป็น method พิเศษ หรือ เหตุการณ์ที่เกิดขึ้นอัตโนมัติเมื่อเพจนี้ถูกเรียกขึ้นมา (enter)
  //เรียกว่า Lifecycle events อ่านเพิ่มเติมได้ที่
https://ionicframework.com/docs/api/navigation/NavController/
  ionViewWillEnter() {
    this.getCourses(); //เรียกใช้ method getCourses()
  }

  ionViewWillLeave() {
    this.sub.unsubscribe(); // unsubscribe ข้อมูลที่มาจากการดึง server เพื่อคืนหน่วยความจำ
  }
}

```

8. เปิดไฟล์ src\pages\course\course.html เพื่อเขียนโค้ดแสดงผลข้อมูลโดยใช้ Lists Component ดังนี้

```

<ion-header>
  <ion-navbar>
    <button ion-button menuToggle>
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-title>รายการคอร์สเรียน</ion-title>
  </ion-navbar>
</ion-header>

<ion-content padding>

  <ion-searchbar (ionInput)="getItems($event)"></ion-searchbar>

  <ion-list>
    <button ion-item *ngFor="let c of courses" (click)="itemSelected(c)">
      <h3>{{ c.c_title }}</h3>
      <p>{{ c.c_detail }}</p>
      <ion-badge item-end color="danger">{{ c.c_view }}</ion-badge>
    </button>
  </ion-list>

</ion-content>

```

ภาพแสดงการผูกข้อมูล และการวนลูปเพื่อแสดงผลข้อมูล

```

course.html
1  <ion-header>
2    <ion-navbar>
3      <button ion-button menuToggle>
4        <ion-icon name="menu"></ion-icon>
5      </button>
6      <ion-title>คอร์สเรียนฟรี</ion-title>
7    </ion-navbar>
8  </ion-header>
9  <ion-content padding>
10    <ion-list>
11      <button ion-item *ngFor="let c of courses">
12        <h3>{{ c.c_title }}</h3>
13        <p>{{ c.c_detail }}</p>
14        <ion-badge item-right color="danger">
15        </ion-badge>
16      </button>
17    </ion-list>
18  </ion-content>

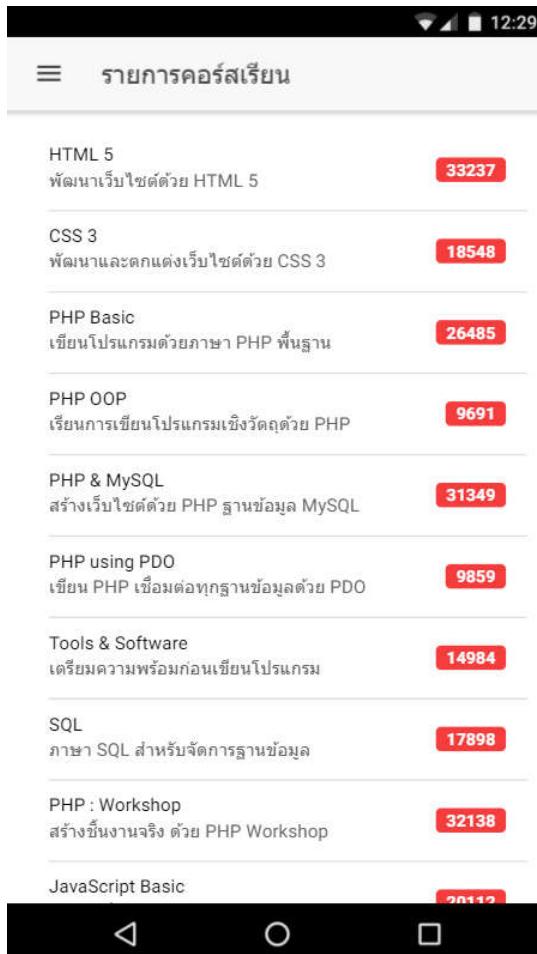
```

```

course.ts
3
4  import { CourseServiceProvider } from '../../../../../';
5  import { Course } from '../../../../../models/course';
6
7  import { Subscription } from 'rxjs/Subscription';
8
9  @IonicPage()
10 @Component({
11   selector: 'page-course',
12   templateUrl: 'course.html'
13 })
14 export class CoursePage {
15   courses: Course[];
16   sub: Subscription;
17   errorMessage: string;
18
19   constructor(public navCtrl: NavController,

```

9. บันทึกไฟล์ทั้งหมด และคัดลอกเพียงเท่านี้ เรายังสามารถดึงข้อมูลจาก Backend ได้เรียบร้อย 😊



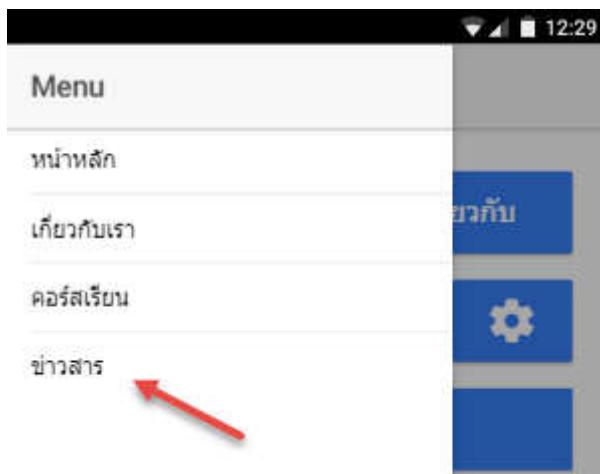
ตอนนี้เรามีความรู้ และสามารถดึงข้อมูลจาก Backend กันได้เรียบร้อย เพื่อทบทวนเราจะมาทำ Workshop กันต่อ โดยจะประยุกต์เพิ่มอีก โดยใช้ Ionic Component อีก ด้วย เช่น Cards, Loading และจะลองแสดงภาพจาก API ด้วย

ใน Workshop นี้เราจะดึงข้อมูลข่าวจากเว็บ <https://newsapi.org/> ซึ่งเค้ามี API ให้เราใช้ด้วย แนะนำให้สมัครสมาชิกก่อน หากไม่อยากสมัครสมาชิก จะลองใช้ apiKey เดียวกันกับในหนังสือเล่มนี้ก็ได้ ตาม URL นี้

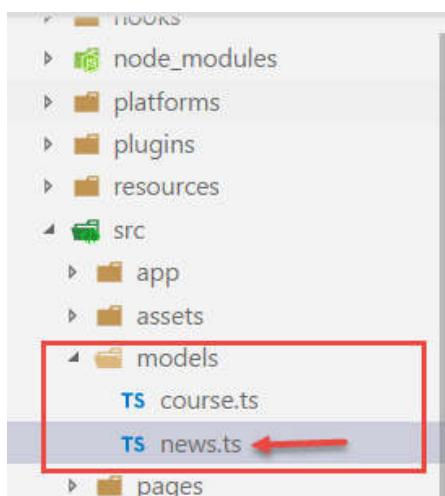
<https://newsapi.org/v2/top-headlines?country=us&apiKey=ab0d4aca4cea481e8157d31c68eb2b23>

หากลองดูข้อมูลจาก URL ด้านบนนี้จะสังเกตว่าข้อมูล JSON ที่ได้จะมีการซ้อนกันอยู่ เพราะฉะนั้นเราทำ Model จะต้องเขียน Model ให้ตรงกับข้อมูล JSON ที่ได้จาก Backend ด้วย มาดูกันเลย!

- ให้สร้างเพจใหม่ชื่อว่า news ด้วยคำสั่ง ionic g page news และกด enter
- ให้ทดลอง import NewsPageModule ในไฟล์ app.module.ts ด้วยตัวเอง
- สร้างเมนูด้านข้าง (ให้ทดลองเขียนเพิ่มเมนูด้านข้างด้วยตัวเอง ที่ไฟล์ app.component.ts) ดังรูป



4. สร้าง model เพื่อรับข้อมูลข่าวที่ได้จาก Backend โดยให้สร้างไฟล์ใหม่ชื่อว่า news.ts ในโฟลเดอร์ src\models ดังนี้



เปิดไฟล์ src\models\news.ts และพิมพ์ได้ดังนี้

```
export interface News {
  status: string;
  totalResults: number;
  articles: Article[];
}

export interface Article {
  source: Source;
  author?: string;
  title: string;
  description: string;
  url: string;
  urlToImage?: string;
  publishedAt: string;
}

export interface Source {
  id?: string;
```

```

    name: string;
}

```

สังเกตว่า ใน interface News จะมี Article[] (ข้าง) ข้อมูลด้านใน และใน Article จะมี Source ข้อมูลเช่นเดียวกัน

- สร้าง provider ใหม่ ด้วยคำสั่ง ionic g provider newsService แล้วกด enter

### **ionic g provider newsService**

- เปิดไฟล์ src\providers\news-service\news-service.ts เพื่อเพิ่มโค้ดสำหรับดึงข้อมูลข่าวจาก Backend ดังนี้

```

import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';

import { News } from '../../models/news';
import { Observable } from 'rxjs/Observable';

@Injectable()
export class NewsServiceProvider {
  apiUrl:string = 'https://newsapi.org/v2/top-headlines?country=us&apiKey=ab0d4aca4cea481e8157d31c68eb2b23';

  constructor(public http: HttpClient) { }

  getNews(): Observable<News> {
    return this.http.get<News>(this.apiUrl);
  }
}

```

- เมื่อสร้าง provider หรือ Service เรียบร้อยแล้ว อย่าลืมตรวจสอบที่ไฟล์ src\app\app.module.ts ด้วยว่ามีการ import และกำหนดในส่วนของ NewsServiceProvider แล้วหรือยัง (ปกติ ionic cli จะเพิ่มให้อัตโนมัติอยู่แล้ว)

- เรียกใช้ provider ให้เปิดไฟล์ src\pages\news\news.ts เปลี่ยนคำสั่ง ดังนี้

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';

import { NewsServiceProvider } from '../../providers/news-service/news-service';
import { News, Article } from '../../models/news';

@IonicPage()

```

```

@Component({
  selector: 'page-news',
  templateUrl: 'news.html',
})
export class NewsPage {
  news: News;
  articles: Article[];
  total: number;

  constructor(public newsServiceProvider:NewsServiceProvider ,public navCtrl: NavController, public navParams: NavParams) {
  }

  ionViewWillEnter() {
    this.getNews();
  }

  getNews(): void {
    this.newsServiceProvider.getNews().subscribe(
      (news) => {
        this.news = news;
        this.total = this.news.totalResults; //គឺជាការណែនាំថាដំឡើងអំពីតម្លៃទូទៅ

        this.articles = news.articles; //តើកើតឡើងពេលថាជាដំឡើងអំពីតម្លៃទូទៅ

      }
    );
  }
}

```

9. เปิดไฟล์ src\pages\news\news.html เพื่อแสดงผลข้อมูลโดยใช้ Cards Component

```

<ion-header>
  <ion-navbar>
    <button ion-button menuToggle>
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-title>ថាមសាន្ត {{ total }} ថាម</ion-title>
  </ion-navbar>
</ion-header>

<ion-content padding>

  <ion-card *ngFor="let n of articles">
    <img *ngIf="n.urlToImage" [src]="n.urlToImage">

      <ion-card-content>
        <ion-card-title>

```

```

        {{ n.title }}
      </ion-card-title>
    <p>
      {{ n.description }}
    </p>
  </ion-card-content>

  <ion-row no-padding>
    <ion-col>
      <button ion-button clear small color="danger" icon-left>
        <ion-icon name='person'></ion-icon>
        {{ n.source.name }}
      </button>
    </ion-col>
    <ion-col text-right>
      <button ion-button clear small color="danger" icon-left>
        <ion-icon name='calendar'></ion-icon>
        {{ n.publishedAt | date:'dd/M/yyyy' }}
      </button>
    </ion-col>
  </ion-row>
</ion-card>

</ion-content>

```

อธิบายโดยดี: เจ้าใช้ \*ngFor เพื่อวนลูป articles (ข่าว) ซึ่งเป็น array ที่เก็บข้อมูลข่าวทั้งหมด ในหัวข้อนี้เราจะได้เรียนรู้การผูกข้อมูล URL ของรูปภาพด้วย นั่นคือ คำสั่ง <img [src]="n.urlToImage"> และการจัดรูปแบบของวันที่ ด้วย Pipe นั่นคือโดยดีนี้ {{ n.publishedAt | date:'dd/M/yyyy' }}

10. ต่อไปเราจะทำสถานะโหลดข้อมูล โดยใช้ Loading Component เปิดไฟล์ src\pages\news\news.ts เพิ่มคำสั่งดังนี้ (ดูโดยดีที่เป็นตัวหนา)

```

import { Component } from '@angular/core';

import { IonicPage, NavController, NavParams, LoadingController } from 'ionic-angular';

import { NewsServiceProvider } from '../../providers/news-service/news-service';

import { News, Article } from '../../models/news';

@IonicPage()
@Component({
  selector: 'page-news',

```

```

templateUrl: 'news.html',
})

export class NewsPage {
  news: News;
  articles: Article[];
  total: number;

  constructor(private loadingCtrl:LoadingController, public newsServiceProvider:NewsServiceProvider ,public navCtrl: NavController, public navParams: NavParams) {
  }

  ionViewWillEnter() {
    this.getNews();
  }

  getNews(): void {
    let loading = this.loadingCtrl.create({
      content: 'ກຳລັງໂຫລດຂໍອມູລ...',  

      //spinner: 'dots'  

    });
    loading.present(); //ເຮີມແສດງ Loading

    this.newsServiceProvider.getNews().subscribe(  

      (news) => {  

        this.news = news;  

        this.total = news.totalResults;//ດຶງຄ່າຈຳນວນຂໍ້າວທັງໝາດ  

        this.articles = news.articles; //ເລືອກຂໍ້ອມູລເຊີພະຂໍ້າວທັງໝາດ  

      },  

      (error) => {  

        // console.log(error);  

        loading.dismiss() //ໃຫ້ Loading ພາຍໄປກຣົນີເກີດ error  

      },
    );
  }
}

```

```

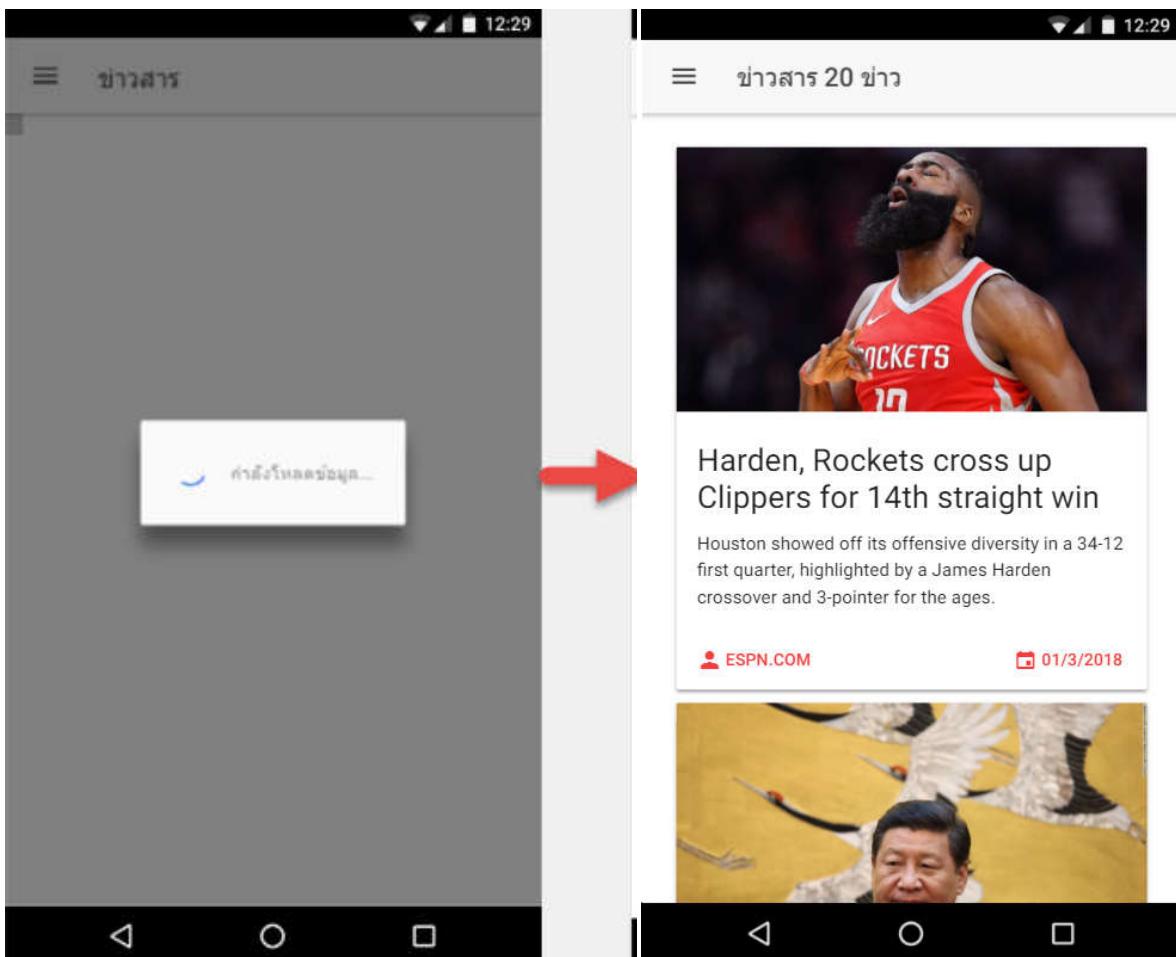
() => loading.dismiss() //ให้ Loading หายไปกรณีเกิดการทำงานเสร็จสมบูรณ์
);
}

}

```

Note: ดูคู่มือการใช้งาน loading เพิ่มเติม ได้ที่ <https://ionicframework.com/docs/components/#loading>

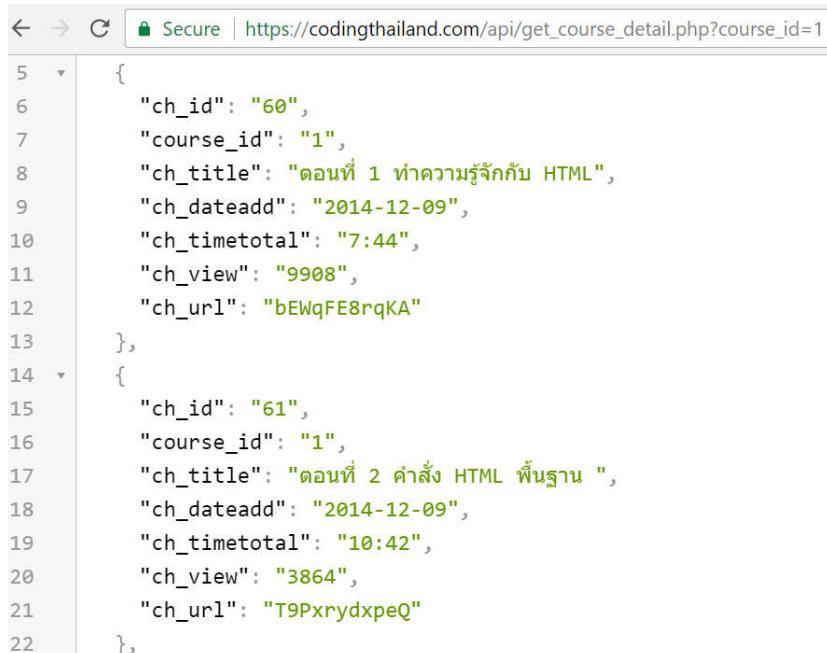
- เพียงเท่านี้เว็บสามารถดึงข้อมูลจาก Backend พร้อมทั้งแสดงสถานะระหว่างรอโหลดข้อมูลเซิร์ฟเวอร์อย่างง่ายดาย 😊



## การสร้าง Pages ในรูปแบบ Master Detail

ในหัวข้อที่แล้วเราสร้างได้เพจ ชื่อว่า course เพื่อโหลดข้อมูลแสดงรายการคอร์สเรียนพรีจากเว็บ codingthailand.com เรียบร้อยแล้ว ในหัวข้อนี้เราจะมาทดลองสร้างเพจในรูปแบบของ Master Detail กัน โดยจะให้เพจ course เป็น Master ที่แสดงคอร์สเรียนทั้งหมด และจะสร้างเพจใหม่ชื่อว่า course-detail สำหรับแสดงรายละเอียดของคอร์สนั้นๆ คือ เมื่อผู้ใช้คลิกเลือกรายการคอร์สใน Lists Component ก็จะแสดงรายละเอียดของแต่ละคอร์สนั้นเอง

สำหรับเพจ course-detail นี้เราจะใช้ Backend ตาม URL [https://codingthailand.com/api/get\\_course\\_detail.php?course\\_id=1](https://codingthailand.com/api/get_course_detail.php?course_id=1)



```

5  {
6      "ch_id": "60",
7      "course_id": "1",
8      "ch_title": "ตอนที่ 1 ทำความรู้จักกับ HTML",
9      "ch_dateadd": "2014-12-09",
10     "ch_timetotal": "7:44",
11     "ch_view": "9908",
12     "ch_url": "bEWqFE8rqKA"
13 },
14 {
15     "ch_id": "61",
16     "course_id": "1",
17     "ch_title": "ตอนที่ 2 คำสั่ง HTML พื้นฐาน",
18     "ch_dateadd": "2014-12-09",
19     "ch_timetotal": "10:42",
20     "ch_view": "3864",
21     "ch_url": "T9PxrydxpeQ"
22 }

```

ตัวเลขด้านหลังสุดของ URL (?course\_id=1) คือ ตัวเลขรหัสคอร์สที่จะต้องรับจากมาจากการ courses จะทดลองเปลี่ยนตัวเลขก็ได้ เช่นเปลี่ยนจาก 1 เป็น 2 เป็นต้น มาลุยกันเลย!

- ให้สร้างเพจใหม่ชื่อว่า course-detail ด้วยคำสั่ง ionic g page course-detail และกด enter (อย่าลืม ทุกครั้งที่สร้างเพจ จะต้อง import CourseDetailPageModule ที่ไฟล์ src\app\app.module.ts ด้วย)
- ก่อนจะไปที่เพจ course-detail เราจะเขียนในส่วนของเพจ Master ก่อน นั่นคือเพจ course ให้เปิดไฟล์ src\pages\course\course.html เพื่อเขียนโค้ดเหตุการณ์คลิก (click) ให้กับปุ่ม เมื่อผู้ใช้คลิกรายการ์สให้เรียกฟังก์ชันที่ชื่อว่า itemSelected พร้อมทั้งส่งตัวแปร c ไปด้วย (c คือ ตัวแปรคอร์สแต่ละคอร์สที่ถูกส่งพร้อมกับข้อมูลไปกับฟังก์ชัน) ดังนี้ (click)="itemSelected(c)"

```

<ion-header>
  <ion-navbar>
    <button ion-button menuToggle>
      <ion-icon name="menu"></ion-icon>

```

```

        </button>
        <ion-title>รายการคอร์สเรียน</ion-title>
    </ion-navbar>
</ion-header>

<ion-content padding>

    <ion-list>
        <button ion-item *ngFor="let c of courses" (click)="itemSelected(c)">
            <h3>{{ c.c_title }}</h3>
            <p>{{ c.c_detail }}</p>
            <ion-badge item-end color="danger">{{ c.c_view }}</ion-badge>
        </button>
    </ion-list>

</ion-content>

```

3. เปิดไฟล์ src\pages\course\course.ts เขียน method itemSelected พั้นทั้งสิ่งค่าตัวแปร c ไปยังหน้า course-detail การส่งตัวแปร c นั้นจะใช้หลักการการส่งข้อมูลระหว่างเพจนั้นเอง

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';

import { CourseServiceProvider } from '../../providers/course-service/course-service'; //import service เข้ามาใช้งาน
import { Course } from '../../models/course'; //import model เข้ามาใช้งาน
import { CourseDetailPage } from './course-detail/course-detail';//import CourseDetailPage

import { Subscription } from 'rxjs/Subscription'; //import Subscription เพื่อ unsubscribe() ข้อมูลจาก Server

@IonicPage()
@Component({
  selector: 'page-course',
  templateUrl: 'course.html'
})
export class CoursePage {
  courses: Course[]; //ประกาศตัวแปร courses มีชนิดข้อมูลเป็นโมเดล Course (array)
  sub:Subscription;
}

```

```

constructor(public navCtrl: NavController,
  public navParams: NavParams,
  private courseServiceProvider:CourseServiceProvider
) {}

//สร้าง method ชื่อว่า getCourses() มีหน้าที่ดึงคอร์สจาก Backend
//subscribe ข้อมูลที่ได้รับจาก service (CourseServiceProvider)
//และส่งค่าให้กับตัวแปร courses (courses (res) => this.courses = res)

private getcourses() {
  this.sub = this.courseServiceProvider.getCourse().subscribe(
    (courses:Course[]) => this.courses = courses //courses ทางฝั่งขวา คือ คอร์สที่มาจากฝั่ง Backend นั่นเอง
  );
}

//ionViewWillEnter() เป็น method พิเศษ หรือ เหตุการณ์ที่เกิดขึ้นอัตโนมัติเมื่อเพจนี้ถูกเรียกขึ้นมา (enter)
//เรียกว่า Lifecycle events อ่านเพิ่มเติมได้ที่ https://ionicframework.com/docs/api/navigation/NavController/
ionViewWillEnter() {
  this.getcourses(); //เรียกใช้ method getcourses()
}

ionViewWillLeave() {
  this.sub.unsubscribe(); // unsubscribe ข้อมูลที่มาจาก server เพื่อคืนหน่วยความจำ
}

itemSelected(c):void {
  this.navCtrl.push(CourseDetailPage,{
    id : c.id, //ส่งรหัสคอร์สที่เลือกพร้อมกำหนดให้กับ id และส่งไปให้หน้า course-detail
    c_detail : c.c_detail //ส่งรายการรายละเอียดคอร์สไปด้วย พร้อมทั้งกำหนดให้กับ title ให้หน้า course-detail
  });
}
}

```

4. ในการแสดงข้อมูลรายละเอียดคอร์ส เราจะใช้วิธีไม่ต้องสร้าง model ลองดู โดยกำหนดชนิดข้อมูลเป็น any แทน
5. เปิดไฟล์ src\providers\course-service\course-service.ts เพื่อเพิ่ม method getCourseDetail สำหรับการดึงข้อมูลรายละเอียดแต่ละคอร์ส เพิ่มได้ดังนี้

```

import { HttpClient } from '@angular/common/http';

import { Injectable } from '@angular/core';

import { Course } from '../../models/course';

import { Observable } from 'rxjs/Observable';

@Injectable()
export class CourseServiceProvider {

  apiUrl:string = 'https://codingthailand.com/api/get_courses.php';
  apiUrl2:string = 'https://codingthailand.com/api/get_course_detail.php';

  constructor(public http: HttpClient) {}

  //เขียน method ชื่อว่า getCourse() เพื่อสำหรับดึงข้อมูลคอร์สวิธกัน
  //คำสั่ง this.http.get<Course[]>(this.apiUrl); คือคำสั่งให้ดึงข้อมูลจาก Backend ด้วย method get() ตาม URL ที่ได้ประกาศไว้ด้านบน
  //และทำการแปลง json จากที่ได้จาก backend ให้กับโมเดล Course
  //จากนั้นก็ return ข้อมูลคอร์สวิธกันทั้งหมดออกไปกับโมเดล Course ในรูปแบบ Observable
  //ทำให้เพจต่างๆสามารถเรียกใช้เมื่อต้อง getCourse() ของ Provider นี้ได้
  getCourse(): Observable<Course[]> {
    return this.http.get<Course[]>(this.apiUrl);
  }

  //ดึงข้อมูลรายละเอียดคอร์สตามรหัสคอร์สที่ส่งเข้ามา
  //เราจะไม่ใช้ model แต่กำหนดเป็น any แทน
  getCourseDetail(id: number): Observable<any[]> {
    // get_course_detail.php?course_id=1
  }
}

```

```

    const param = { 'course_id': id.toString() };

    return this.http.get< any[]>(this.apiUrl2, {params: param});

}

}

```

6. เปิดไฟล์ src\pages\course-detail\course-detail.ts เพื่อรับค่าที่ส่งมาจากการเพจ course พร้อมทั้งโหลดหัวข้อรายละเอียดคอร์สต่างๆ

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';

import { CourseServiceProvider } from '../../providers/course-
service/course-service';

@IonicPage()
@Component({
  selector: 'page-course-detail',
  templateUrl: 'course-detail.html',
})
export class CourseDetailPage {
  id:number;
  c_detail:string;
  courseDetail: Array<any>; //ประเภท array แบบ generic

  constructor(public courseServiceProvider:CourseServiceProvider, public
navCtrl: NavController, public navParams: NavParams) {
    this.id = this.navParams.get('id');
    this.c_detail = this.navParams.get('c_detail');
  }

  ionViewWillEnter() {
    this.courseServiceProvider.getCourseDetail(this.id).subscribe(
      (courseDetail) => this.courseDetail = courseDetail
    );
  }
}

```

7. เปิดไฟล์ src\pages\course-detail\course-detail.html เพื่อเขียนโค้ดแสดงผลหัวข้อมูลโดยใช้ Lists Component ดังนี้

```

<ion-header>
  <ion-navbar>
    <ion-title>{{ c_detail }}</ion-title>
  </ion-navbar>
</ion-header>

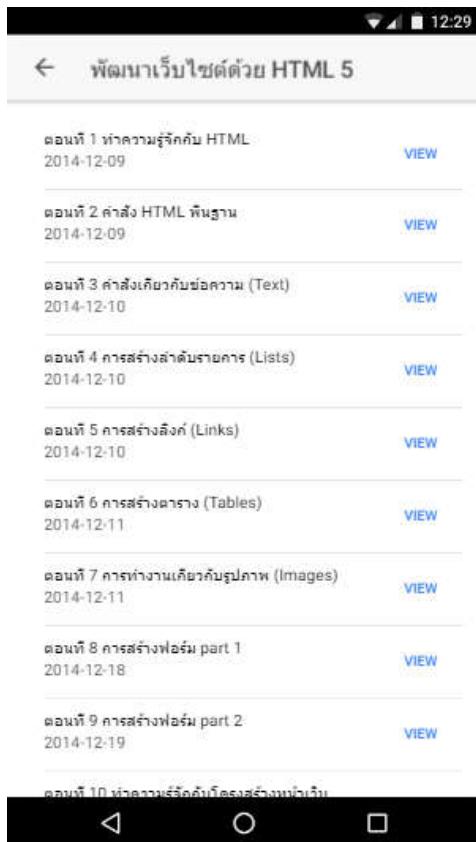
```

```

<ion-content padding>
  <ion-list>
    <button ion-item *ngFor="let c of courseDetail"
(click)="itemSelected(c)">
      <h3>{{ c.ch_title }}</h3>
      <p>{{ c.ch_dateadd }}</p>
      <button ion-button clear item-right>View</button>
    </button>
  </ion-list>
</ion-content>

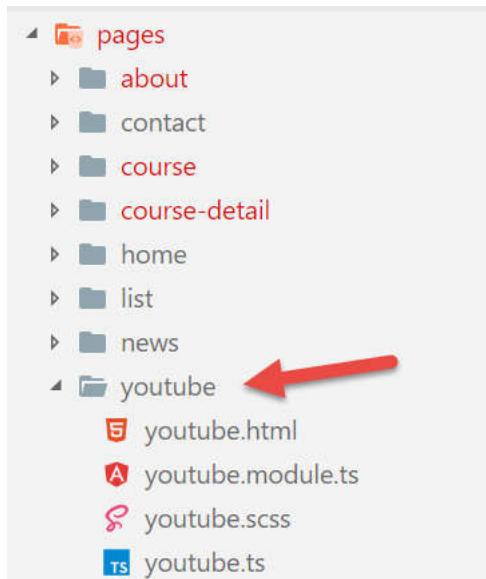
```

8. บันทึกไฟล์ แล้ววันเพื่อทดสอบ App โดยลองกดเลือกที่รายการคอร์สเรียน เมื่อกดแล้วจะเห็นหัวข้อรายละเอียดการสอนในคอร์สนี้ๆ



ถึงตรงนี้แล้วเราจะมาลองสร้างอีก 1 เพจเพื่อให้แสดงวิดีโอจาก Youtube กัน คือเมื่อคลิกที่รายละเอียดหัวข้อเรียนแล้วให้เปิดหน้าเพจ youtube ขึ้นมาแสดง แต่! ในหัวข้อนี้จะให้ทำแบบฝึกหัดในการสร้างเพจเอง และจะให้ขั้นตอนไว้เช่นๆ มาเริ่มกันเลย

- สร้างเพจชื่อว่า youtube (ให้ทดลองสร้างด้วยตัวเอง)



- เปิดไฟล์ src\pages\course-detail\course-detail.ts เพื่อส่งค่า URL ของ youtube ให้เพจ youtube

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';

import { CourseServiceProvider } from '../../providers/course-service/course-service';
import { YoutubePage } from './youtube/youtube'; //import Youtube page

@IonicPage()
@Component({
  selector: 'page-course-detail',
  templateUrl: 'course-detail.html',
})
export class CourseDetailPage {
  id:number;
  c_detail:string;
  courseDetail: Array<any>; //ประกาศ array แบบ generic

  constructor(public courseServiceProvider:CourseServiceProvider, public navCtrl: NavController, public navParams: NavParams) {
  }
}

```

```

this.id = this.navParams.get('id');

this.c_detail = this.navParams.get('c_detail');

}

ionViewWillEnter() {

  this.courseServiceProvider.getCourseDetail(this.id).subscribe(
    (courseDetail) => this.courseDetail = courseDetail
  );
}

itemSelected(c):void {
  this.navCtrl.push(YoutubePage,{
    ch_title : c.ch_title, //ส่งรายละเอียดหัวข้อคอร์สไปให้เพจ Youtube
    ch_url : c.ch_url //ส่ง Youtube id ไปให้เพจ Youtube เพื่อนำไปแสดงวิดีโอ
  });
}
}

```

3. เปิดไฟล์ src\pages\youtube\youtube.ts เพื่อทำการรับค่าที่ส่งมาจากเพจ course-detail เรียนรู้ดังนี้

```

import { Component } from '@angular/core';

import { IonicPage, NavController, NavParams } from 'ionic-angular';

import { DomSanitizer, SafeResourceUrl } from '@angular/platform-browser';

@IonicPage()
@Component({
  selector: 'page-youtube',
  templateUrl: 'youtube.html'
})
export class YoutubePage {
  chTitle:string;

```

```

chUrl:string;
chUrlTrusted: SafeResourceUrl;

constructor(private domSanitizer:DomSanitizer ,public navCtrl: NavController, public navParams: NavParams) {
  this.chTitle = this.navParams.get('ch_title');
  this.chUrl = 'https://www.youtube.com/embed/'+this.navParams.get('ch_url');

}

ionViewWillEnter() {
  this.chUrlTrusted = this.domSanitizer.bypassSecurityTrustResourceUrl(this.chUrl);
}

}

```

Note: Angular เข้มงวดเรื่องความปลอดภัย ในเพจนี้เราใช้ <iframe> ซึ่งเป็นการฝังโค้ด และถูกมองว่าไม่ปลอดภัย ดังนั้นเราจึงต้องอนุญาต URL Youtube นี้ ด้วยการเขียนโค้ดโดยใช้ method bypassSecurityTrustResourceUrl() ก่อน 以便เมื่อเรื่องความปลอดภัยได้ที่นี่ <https://angular.io/guide/security>

4. เปิดไฟล์ src\pages\youtube\youtube.html เรียนโค้ดเพื่อแสดงวิดีโอ และรายละเอียดหัวข้อที่สอน ดังนี้

```

<ion-header>
  <ion-navbar>
    <ion-title>{{ chTitle }}</ion-title>
  </ion-navbar>
</ion-header>

<ion-content padding>

  <iframe *ngIf="chUrlTrusted" [src]="chUrlTrusted" style="width: 100%; height: 100%"></iframe>
  <button ion-button full navPop>กลับ</button>

</ion-content>

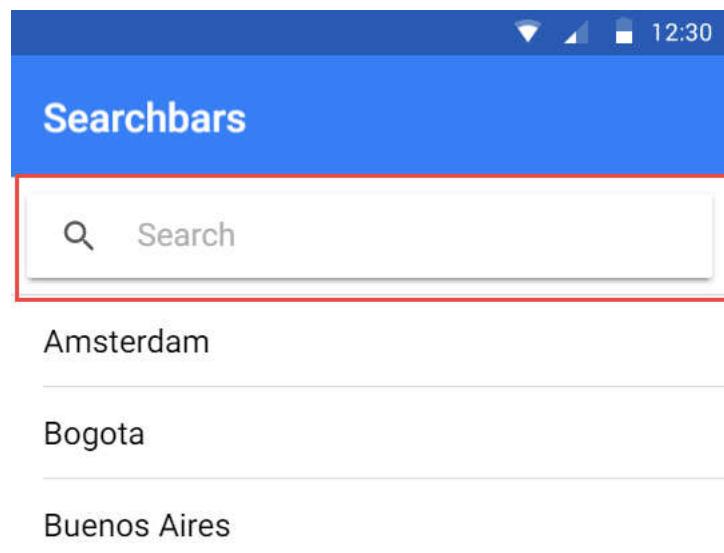
```

5. บันทึกไฟล์ ทดลองรัน App ทดสอบโดยกดเลือกที่หัวข้อการสอนต่างๆ



#### การใช้ Component Searchbar สำหรับค้นหาข้อมูล

หลังจากที่เราสร้าง Lists Component กันมาพอสมควร หากเราต้องการค้นหาข้อมูล หรือการองข้อมูล แนะนำให้ใช้ Searchbar ดังรูป



Note: รายละเอียด Searchbar ดูได้ที่ <https://ionicframework.com/docs/components/#searchbar>

เมื่อผู้ใช้ค้นข้อมูล เราจะต้องใช้ฟังก์ชันของ JavaScript เข้ามาช่วยในการค้นหา เช่น indexOf, filter, includes เป็นต้น การใช้งานฟังก์ชันพวgnนี้อยู่ในความรู้พื้นฐานของ JavaScript อยู่แล้ว ในตัวอย่างนี้เราจะลองทำ Searchbar เพื่อค้นหาข้อมูลคอร์สเรียนกัน 黎มเลย!

1. เปิดไฟล์ src\pages\course\course.html เพื่อแทรก Searchbar Component เข้าไป (ตัวหนา) ดังนี้

```
<ion-header>
  <ion-navbar>
    <button ion-button menuToggle>
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-title>รายการคอร์สเรียน</ion-title>
  </ion-navbar>
</ion-header>

<ion-content padding>

  <ion-searchbar (ionInput)="getItems($event)"></ion-searchbar>

  <ion-list>
    <button ion-item *ngFor="let c of courses"
(click)="itemSelected(c)">
      <h3>{{ c.c_title }}</h3>
      <p>{{ c.c_detail }}</p>
      <ion-badge item-end color="danger">{{ c.c_view }}</ion-
badge>
    </button>
  </ion-list>

</ion-content>
```

2. เปิดไฟล์ src\pages\course\course.ts สร้าง method getItems() เพื่อช่วยค้นหาข้อมูล ดังนี้

```
import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';

import { CourseServiceProvider } from '../../providers/course-
service/course-service'; //import service เข้ามาใช้งาน
import { Course } from '../../models/course'; //import model เข้ามาใช้งาน
import { CourseDetailPage } from '../course-detail/course-
detail';//import CourseDetailPage
```

```

import { Subscription } from 'rxjs/Subscription'; //import Subscription
เพื่อ unsubscribe() ข้อมูลจาก Server

@IonicPage()
@Component({
  selector: 'page-course',
  templateUrl: 'course.html'
})
export class CoursePage {
  courses: Course[]; //ประการตัวแปร courses มีชนิดข้อมูลเป็นโมเดล Course (array)
  sub: Subscription;

  constructor(public navCtrl: NavController,
              public navParams: NavParams,
              private courseServiceProvider: CourseServiceProvider
            ) {}

  //สร้าง method ชื่อว่า getCourse() มีหน้าที่คิ่งคอร์สจาก Backend
  //subscribe ข้อมูลที่ได้รับจาก service (CourseServiceProvider)
  //และส่งค่าให้กับตัวแปร courses (courses (res) => this.courses = res)
  private getCourses() {
    this.sub = this.courseServiceProvider.getCourse().subscribe(
      (courses: Course[]) => this.courses = courses //courses ทางซ้ายขวาคือ คอร์ส
      ที่มาจากการ Backend นั้นเอง
    );
  }

  //ionViewWillEnter() เป็น method พิเศษ หรือ เหตุการณ์ที่เกิดขึ้นอัตโนมัติเมื่อเจอกับหน้า (enter)
  //เรียกว่า Lifecycle events อ่านเพิ่มเติมได้ที่
https://ionicframework.com/docs/api/navigation/NavController/
  ionViewWillEnter() {
    this.getCourses(); //ใช้ method getCourses()
  }

  ionViewWillLeave() {
    this.sub.unsubscribe(); // unsubscribe ข้อมูลที่มาจาก server เพื่อคืนหน่วยความจำ
  }

  itemSelected(c): void {
    this.navCtrl.push(CourseDetailPage, {
      id: c.id, //ส่งรหัสอโตรส์ที่เลือกพร้อมกำหนดให้กับ id แล้วส่งไปให้หน้า course-detail
      c_detail: c.c_detail //ส่งรายการรายละเอียดคอร์สไปด้วย พร้อมทั้งกำหนดให้กับ title ให้หน้า course-detail
    });
  }
}

```

```
//ค้นหา
getItems(ev: any) {
  // set val to the value of the searchbar
  let val = ev.target.value;
  // if the value is an empty string don't filter the items
  if (val && val.trim() != '') {
    this.courses = this.courses.filter((course:Course) => {
      return (course.c_detail.toLowerCase().indexOf(val.toLowerCase()) > -1); //กรณีหากค้นหาด้วย c_detail
    })
  } else {
    this.getCourses(); //ถ้าห้องค้นหาว่างให้โหลดข้อมูลทั้งหมดอีกครั้ง
  }
}
}
```

### 3. ทดสอบบันทึก App และลองพิมพ์ค้นหา

ชื่อคอร์สเรียน	รายละเอียด	จำนวนผู้เรียน
PHP Basic เขียนโปรแกรมด้วยภาษา PHP พื้นฐาน	เขียนโปรแกรมด้วยภาษา PHP พื้นฐาน	17040
PHP OOP เรียนการเขียนโปรแกรมเชิงวัสดุด้วย PHP	เรียนการเขียนโปรแกรมเชิงวัสดุด้วย PHP	6370
PHP & MySQL สร้างเว็บไซต์ด้วย PHP MySQL	สร้างเว็บไซต์ด้วย PHP MySQL	20491
PHP using PDO เขียน PHP เข้ามือกับ PDO	เขียน PHP เข้ามือกับ PDO	7180
PHP : Workshop สร้างชีวิตงานจริง ด้วย PHP Workshop	สร้างชีวิตงานจริง ด้วย PHP Workshop	23917
PHP AJAX Real world PHP AJAX and Report	Real world PHP AJAX and Report	1662

## บทที่ 8 การสร้างฟอร์ม (Form)

ในการสร้าง Mobile App มีหลายครั้งที่เราต้องให้ผู้ใช้กรอกข้อมูลผ่านฟอร์ม เช่น ฟอร์มสมัครสมาชิก ฟอร์มล็อกอิน ฟอร์มเพิ่มข้อมูลต่างๆ เป็นต้น ใน Ionic นั้น การสร้างฟอร์มต่างๆ จะใช้ความสามารถของ Angular หากอยากรู้ว่าเพิ่มเติมให้ค้นเรื่อง Angular Form ได้ การเขียนฟอร์มใน Angular นั้น มี 2 รูปแบบ ได้แก่

1. Template driven forms (เขียน markup ที่ template อย่างเดียว)

- ngModel
- NgForm

2. Model driven or reactive forms (เขียนโค้ด markup ที่ template และต้องเขียนที่ส่วนคลาสด้วย)

- formControl
- ngFormGroup

Note: ดูรายละเอียดของ Angular Form ได้ที่ <https://angular.io/docs/ts/latest/guide/forms.html>

### ลองสร้างฟอร์มสมัครสมาชิก (ลงทะเบียน)

ในหัวข้อนี้เราจะมาสร้างฟอร์มสมัครสมาชิก เมื่อผู้ใช้กรอกข้อมูลแล้วจะไปเพิ่มข้อมูลที่ Backend และจะให้แสดง Alert เมื่อบันทึกข้อมูล เรียบร้อยแล้ว เราจะใช้ URL [https://codingthailand.com/api/insert\\_user.php](https://codingthailand.com/api/insert_user.php) เพื่อให้สำหรับเพิ่มข้อมูลลงในตารางผู้ใช้ และจะใช้ URL [https://codingthailand.com/api/get\\_user.php](https://codingthailand.com/api/get_user.php) สำหรับดึงข้อมูลผู้ใช้ที่ได้เพิ่มลงไว้ ในไฟล์ insert\_user.php มีโค้ดตัวอย่าง ดังนี้

```
<?php

header('Access-Control-Allow-Origin: *');

header('Access-Control-Allow-Methods: GET, POST, PATCH, PUT, DELETE, OPTIONS');

header('Access-Control-Allow-Headers: Origin, Content-Type, X-Auth-Token');

$link = mysqli_connect('localhost', 'root', '', 'mydb');

mysqli_set_charset($link, 'utf8');

if ($_SERVER['REQUEST_METHOD'] == 'POST') {

    // get post body content

    $content = file_get_contents('php://input');
```

```

// parse JSON

$user = json_decode($content, true);

$fullname = $user['fullname'];
$email = $user['email'];
$password = $user['password'];

//check duplicate $email

$sql2 = "SELECT email FROM user WHERE email='$email' ";
$result2 = mysqli_query($link, $sql2);
$rowcount = mysqli_num_rows($result2);

if ($rowcount == 1) {
    echo json_encode(['status' => 'error', 'message' => 'ມີສາມາດລົງທະບຽນໄດ້ ອືເມລັນເນື້ອໃຫ້ແລ້ວ']);
    exit;
}

//insert data

$sql = "INSERT INTO `user` (`fullname`, `email`, `pwd`) VALUES ('$fullname', '$email', '$password');";

$result = mysqli_query($link, $sql);

if ($result) {
    echo json_encode(['status' => 'ok', 'message' => 'ບັນທຶກຂໍ້ອມູນເຮືອຍປ້ອຍນະ']);
} else {
    echo json_encode(['status' => 'error', 'message' => 'ເກີດຂໍ້ອົດພລາດໃນກາງບັນທຶກຂໍ້ອມູນ']);
}

mysqli_close($link);

```

## ขั้นตอนการสร้างฟอร์ม มีดังนี้

- ให้สร้างเพจใหม่ชื่อว่า signup ด้วยคำสั่ง ionic g page signup และกด enter (อย่าลืม import SignupPageModule ที่ไฟล์ src\app\app.module.ts ด้วย)
- เปิดไฟล์ src\pages\home\home.html ให้แก้ไขเฉพาะส่วนของ <ion-header> เพื่อเพิ่มปุ่มบน Toolbar พร้อมกับผูก event click ดังนี้

```
<ion-header>
  <ion-toolbar>
    <button ion-button icon-only menuToggle>
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-title>หน้าหลัก</ion-title>
    <ion-buttons end>
      <button ion-button icon-only color="royal" (click)="openSignup()">
        <ion-icon name="person-add"></ion-icon>
      </button>
    </ion-buttons>
  </ion-toolbar>
</ion-header>
```

- เปิดไฟล์ src\pages\home\home.ts เพื่อเขียนโค้ดเปิดเพจ signup โดยให้ import เพจ signup และเขียนโค้ด method openSignup() ดังนี้

```
import { Component } from '@angular/core';
import { NavController } from 'ionic-angular';

import { SignupPage } from '../signup/signup';

@Component({
  selector: 'page-home',
  templateUrl: 'home.html'
})
export class HomePage {
```

```

constructor(public navCtrl: NavController) {

}

openSignup() {
  this.navCtrl.push(SignupPage);
}

}

```

4. สร้าง provider ใหม่ ด้วยคำสั่ง ionic g provider authService และกด Enter
5. สร้าง model เพื่อรองรับข้อมูลการตอบกลับ feedback (JSON) จาก Backend (ไฟล์ insert\_user.php) ให้สร้างไฟล์ใหม่ชื่อว่า feedback.ts ในโฟลเดอร์ src\models เขียนโค้ด ดังนี้

```

export interface FeedBack {
  status:string,
  message:string
}

```

6. เปิดไฟล์ src\providers\auth-service\auth-service.ts เพื่อเขียนคำสั่งสำหรับบันทึกข้อมูลไปที่ Backend ดังนี้

```

import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';

import { FeedBack } from '../models/feedback'; //import model
FeedBack สำหรับใช้งาน
import { Observable } from 'rxjs/Observable';

@Injectable()
export class AuthServiceProvider {
  apiUrl:string = 'https://codingthailand.com/api/insert_user.php';

  constructor(public http: HttpClient) {
    console.log('Hello AuthServiceProvider Provider');
  }
}

```

```

//method signup รับพารามิเตอร์ 3 ตัว และคืนค่า FeedBack กลับออกไป
//สังเกตว่า FeedBack จะไม่ใช่ Array เพราะข้อมูล json ที่ส่งกลับมาจาก Server เป็นข้อมูลชุดเดียว
public signup(fullname:string, email:string, password:string): Observable<FeedBack> {
    const header = { 'Content-Type': 'application/json' }; //กำหนด
header เพื่อส่งเป็น json

    //ประกาศตัวแปร data เพื่อกีบข้อมูลที่รับมา
    let data = {
        'fullname': fullname,
        'email': email,
        'password': password
    }
    //ใช้ method post() สำหรับส่ง data เพื่อไปบันทึกข้อมูล
    return this.http.post<FeedBack>(this.apiUrl, data, { headers:
header });
}
}

```

Note: หากเป็นการเพิ่มข้อมูลให้ใช้ method post() แทน get() นั้นเอง

7. เปิดไฟล์ src\pages\signup\signup.html เพื่อสร้างฟอร์มแบบ Template driven forms ให้เขียนโค้ด ดังนี้

```

<ion-header>
  <ion-navbar>
    <ion-title>ลงทะเบียน</ion-title>
  </ion-navbar>
</ion-header>

<ion-content padding>

<form novalidate #myForm="ngForm" (ngSubmit)="signup(myForm.value)">
  <ion-list>
    <ion-item>
      <ion-label floating>ชื่อ-สกุล</ion-label>
      <ion-input ngModel type="text" name="fullname"
required></ion-input>
    </ion-item>

    <ion-item>
      <ion-label floating>Email</ion-label>

```

```

        <ion-input ngModel type="text" name="email" required
pattern="[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,4}$"></ion-input>
      </ion-item>

      <ion-item>
        <ion-label floating>Password</ion-label>
        <ion-input ngModel type="password" name="password" required
minlength="3"></ion-input>
      </ion-item>

      <ion-item>
        <button [disabled]="myForm.invalid" ion-button large full>ສ້າງບັນຫຼຸງ
ຜູ້ໃຊ້ໃໝ່
      </button>
      </ion-item>
    </ion-list>
  </form>

</ion-content>

```

8. เปิดไฟล์ src\pages\signup\signup.ts เพื่อเขียนโค้ดส่งข้อมูลจากที่ได้รับจากฟอร์มส่งให้ provider บันທຶກข้อมูล และในข้อนี้เราจะประยุกต์ใช้งาน Loading Controller และ Alert Controller เพื่อแสดง feedback ที่ได้รับมาจาก Backend ด้วย

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams, LoadingController,
AlertController } from 'ionic-angular';

import { AuthServiceProvider } from '../../providers/auth-service/auth-
service';
import { FeedBack } from '../../models/feedback';

import { HomePage } from '../home/home';

@IonicPage()
@Component({
  selector: 'page-signup',
  templateUrl: 'signup.html',
})
export class SignupPage {
  feedback:FeedBack;

  constructor(
    public navCtrl: NavController,
    public navParams: NavParams,
    private authServiceProvider: AuthServiceProvider,
    private loadingCtrl: LoadingController,
    private alertCtrl: AlertController) {

```

```

}

signup(myForm) {
    //Object ของฟอร์มทั้งหมดสามารถ log ออกมาได้ หรือจะส่งไปทั้ง Object ก็ได้ เช่นเดียวกัน
    //console.log(myForm);
    //รับข้อมูลต่างๆมาจากฟอร์ม
    let fullname = myForm.fullname;
    let email = myForm.email;
    let password = myForm.password;

    //แสดง loading controller
    let loader = this.loadingCtrl.create({
        content: "กำลังบันทึกข้อมูล..."
    });
    loader.present();

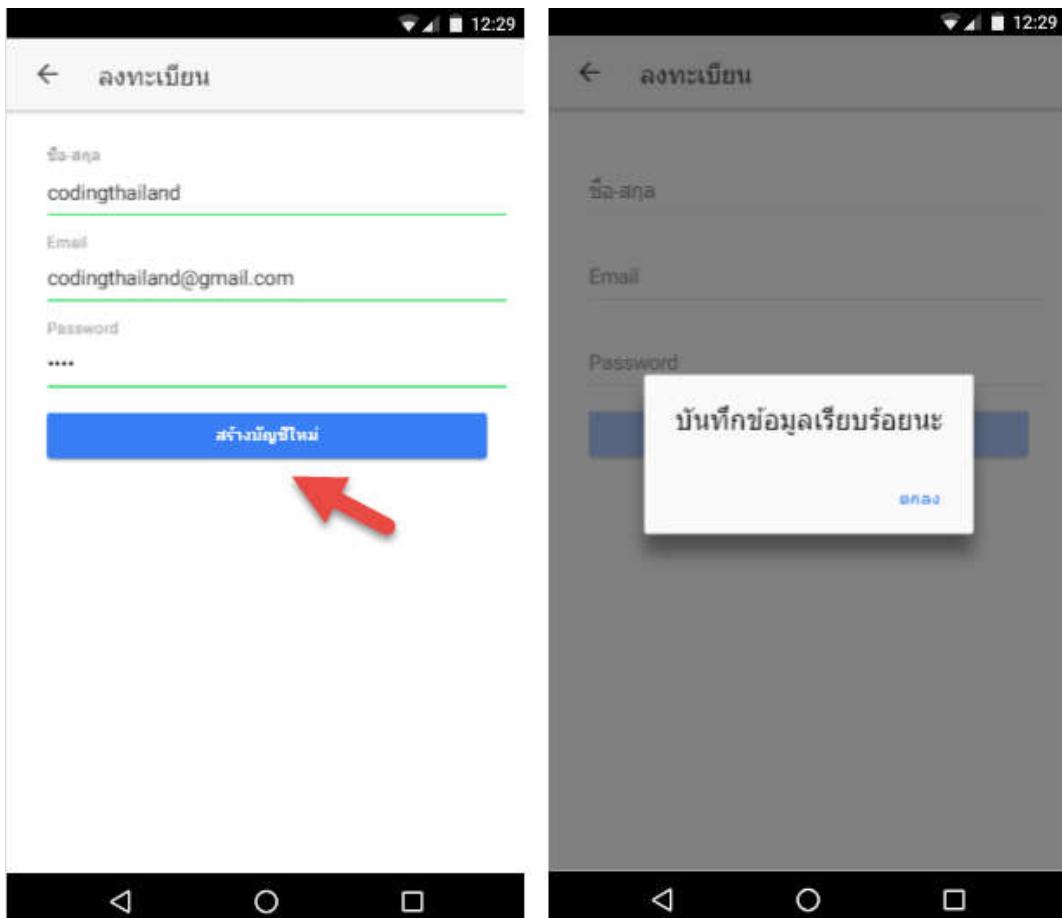
    //ใช้ provider (AuthServiceProvider) และ method signup
    this.authServiceProvider.signup(fullname, email,
password).subscribe(
        (feedback:FeedBack) => {
            this.feedback = feedback; //รับค่าข้อมูล json จาก provider (Backend)
            if (this.feedback.status === 'ok') { //ถ้าสถานะเท่ากับ 'ok' แสดงว่าบันทึก
                //ข้อมูลเรียบร้อย
                let alert = this.alertCtrl.create({
                    title: this.feedback.message,
                    buttons: ['ตกลง']
                });
                //console.log('signup ok');
                alert.present();
                //บันทึกเสร็จเรียบร้อยให้กลับหน้า Home
                this.navCtrl.setRoot(HomePage);

            } else { //ถ้าสถานะเท่ากับ 'error' ให้ทำงานและแสดงข้อความในส่วนนี้
                let alert = this.alertCtrl.create({
                    title: this.feedback.message,
                    buttons: ['ตกลง']
                });
                // console.log('signup not ok');
                alert.present();
            }
        },
        (error) => {
            console.log(error);
            loader.dismiss();
        },
        () => {
            loader.dismiss();
        }
    )
}

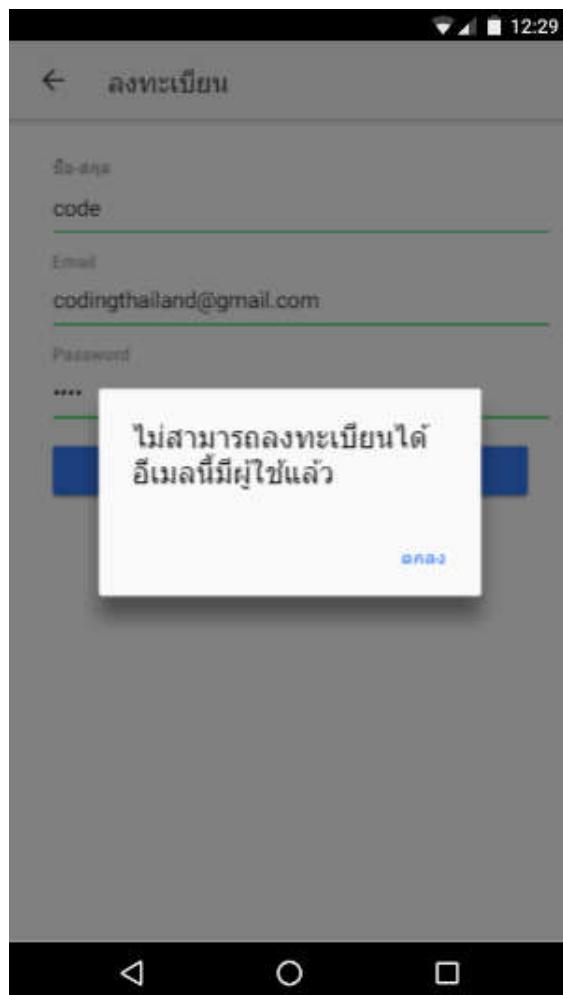
```

```
    );
}
}
```

9. ทดสอบวัน App กรอกข้อมูล แล้วคลิกปุ่มสร้างบัญชีใหม่เพื่อทดสอบ



หากมีการกรอกอีเมลซ้ำ ในไฟล์ insert\_user.php ก็ได้เช่นโดยคิดสำหรับเช็คอีเมลซ้ำไว้แล้ว จะได้ผลลัพธ์ ดังนี้



10. หากต้องการดูข้อมูลที่กรอกไปแล้วสามารถเข้าดูได้ตาม URL [https://codingthailand.com/api/get\\_user.php](https://codingthailand.com/api/get_user.php)



## รายชื่อผู้ใช้ (ใช้เพื่อการเรียนการสอนเท่านั้น)

รหัส	ชื่อ	Username	Password
2	codingthailand	codingthailand@gmail.com	1234
1	ionic 2	ionic2@gmai.com	12345678

## บทที่ 9 การเก็บข้อมูลแบบ Local ด้วย Ionic Storage

หากเราต้องการเก็บข้อมูลแบบ Key/value ไว้ในเครื่องผู้ใช้ หรือต้องการเก็บข้อมูลในรูปแบบของฐานข้อมูล (SQLite) Ionic ได้เตรียมการจัดการตรงนี้ให้เราเรียบร้อยแล้ว เรียกว่า “Ionic Storage” หากเราใช้งานจริง (เครื่องจริง) Ionic จะเลือกฐานข้อมูล SQLite จัดเก็บข้อมูลให้เราเป็นลำดับแรก หากเราใช้ Ionic เขียนเป็น Web App ตัว Ionic Storage จะเลือกใช้ IndexedDB, WebSQL หรือ LocalStorage แทน

สำหรับ SQLite นั้นสามารถเขียนภาษา SQL เพื่อใช้จัดการกับข้อมูลได้ ไม่ว่าจะเป็นคำสั่ง SELECT, INSERT, UPDATE หรือ DELETE เป็นต้น

Note: ดูคู่มือ Ionic Storage ได้ที่ <https://ionicframework.com/docs/storage/>

### การติดตั้ง Ionic Storage

ก่อนจะใช้งาน Ionic Storage เราจะต้องติดตั้ง Cordova Plugin ก่อน (ใช้ SQLite) ด้วยคำสั่ง

```
ionic cordova plugin add cordova-sqlite-storage
```

Note: ขั้นตอนนี้หากเราอย่างไม่เคยติดตั้ง cli-plugin-cordova ก็ให้ตอบ yes ได้เลย

### การใช้งาน Ionic Storage เพื่อเก็บข้อมูลแบบ Key/Value

การเก็บข้อมูลแบบ Key/Value สามารถประยุกต์ใช้เพื่อเก็บข้อมูลได้หลายอย่าง เช่น เอาไว้เก็บ token สำหรับระบบล็อกอิน (จะได้รู้ว่าผู้ใช้ล็อกอินอยู่แล้ว), เก็บการตั้งค่าต่างๆ ใน App ของเรา (แนะนำข้อมูลไม่หาย) เมื่อเก็บข้อมูลแล้วสามารถเรียกใช้ หรือลบโดยการอ้าง Key ได้ตลอดเวลาในทุกๆ เหตุ

### ขั้นตอนการใช้งาน และเก็บข้อมูลแบบ Key/Value

- ก่อนจะใช้งาน Storage จะต้อง import IonicStorageModule ที่ไฟล์ src\app\app.module.ts และต้องกำหนดส่วน imports ด้วย

```

A app.module.ts x
21 import { IonicStorageModule } from '@ionic/storage';
26
27 @NgModule({
28   declarations: [
29     MyApp,
30     HomePage,
31     ListPage
32   ],
33   imports: [
34     BrowserModule,
35     HttpClientModule,
36     IonicModule.forRoot(MyApp),
37     IonicStorageModule.forRoot(), ← 2
38     AboutPageModule, ContactPageModule,
39     CoursePageModule, NewsPageModule, CourseDetailPageModule,
40     YoutubePageModule, SignupPageModule
41   ],

```

2. หากเพจไหนต้องการใช้งาน Storage ก็ให้ import และประกาศที่ constructor ของคลาสก็พร้อมให้งานเรียบร้อย ในหัวข้อนี้จะยกตัวอย่างที่เพจ about ให้เปิดไฟล์ src\pages\about\about.ts เพิ่มโค้ด ดังนี้

```

import { Component } from '@angular/core';

import { IonicPage, NavController, NavParams } from 'ionic-angular';

import { ContactPage } from '../contact/contact';

import { Storage } from '@ionic/storage';

@IonicPage()
@Component({
  selector: 'page-about',
  templateUrl: 'about.html',
})
export class AboutPage {

  constructor(private storage: Storage, public navCtrl: NavController, public navParams: NavParams) {
}

```

```

ionViewWillEnter() {
    this.storage.ready().then(() => { //ถ้า platform พร้อมใช้งาน
        // set a key/value
        //กำหนด และ set ค่า key เป็น phone และค่า value เป็น 085 4952624
        //เราสามารถเรียกใช้ค่านี้ได้จากทุกๆเพจ โดยอ้างชื่อ key นั่นก็คือ phone
        this.storage.set('phone', '085 4952624');

    });
}

goToContact() {
    this.navCtrl.push(ContactPage,{
        companyName: 'CodingThailand',
        companyWebsite: 'https://codingthailand.com'
    });
    //this.navCtrl.setRoot(ContactPage);
}
}

```

3. เมื่อ set ค่าแล้ว เราสามารถนำค่า (value) ข้อมูลมาใช้ได้โดยการอ้าง key (ในที่นี้เราตั้งชื่อว่า phone) เมื่อ set ค่าแล้ว ทุกเพจ สามารถ get ค่ามาใช้งานได้ ในตัวอย่างนี้เราจะใช้เพจ contact เป็นตัวอย่างเพื่อ get ค่าข้อมูลมาแสดง ให้เปิดไฟล์ src\pages\contact\contact.ts เพิ่มโค้ด ดังนี้

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';

import { Storage } from '@ionic/storage'; //import Storage เข้ามาใช้งาน

@IonicPage()
@Component({
    selector: 'page-contact',
    templateUrl: 'contact.html',
})

```

```

        })

export class ContactPage {
    companyName:string;
    companyWebsite:string;
    phone:string;

constructor(private storage:Storage, public navCtrl: NavController, public navParams: NavParams) {
    this.companyName = this.navParams.get('companyName');
    this.companyWebsite = this.navParams.get('companyWebsite');
}

ionViewWillEnter() {
    this.storage.ready().then(() => {
        //get a key/value pair from about page
        //get ค่าข้อมูล key ชื่อว่า phone
        this.storage.get('phone').then((val) => {
            this.phone = val; //ได้ค่าอะไรให้เก็บค่าไว้ให้ตัวแปร phone เพื่อนำไปแสดงผลที่ views
        })
    });
}

ionViewWillLeave() {
    this.storage.remove('phone');
}
}

```

4. เปิดไฟล์ src\pages\contact\contact.html เพื่อผูกข้อมูลกับตัวแปร phone และแสดงเป็นรูปภาพที่ เมื่อคลิก ดังนี้

```

<ion-header>
<ion-navbar>
    <button ion-button menuToggle>
        <ion-icon name="menu"></ion-icon>
    </button>

```

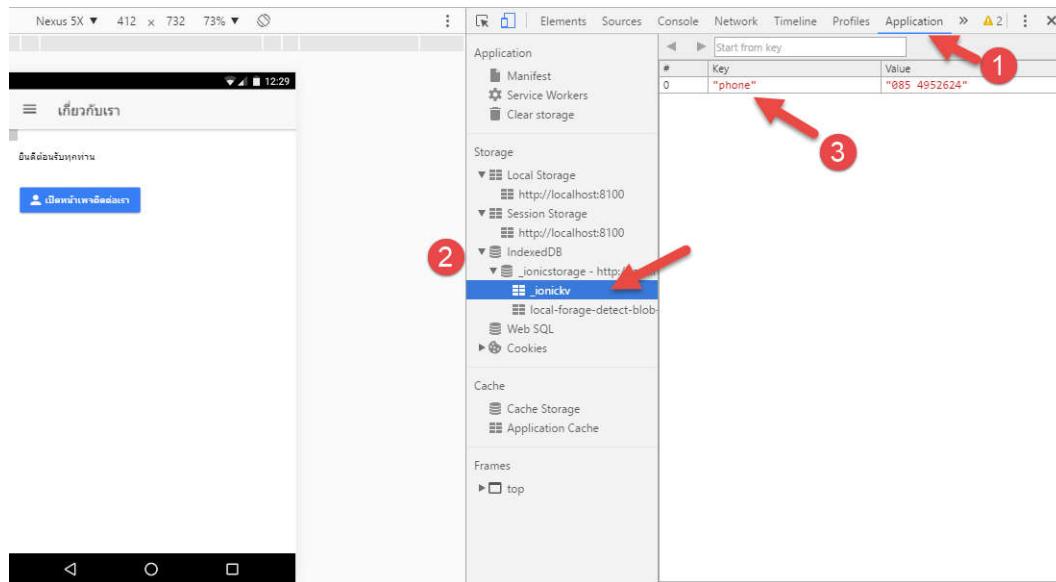
```

<ion-title>ติดต่อเรา</ion-title>
</ion-navbar>
</ion-header>

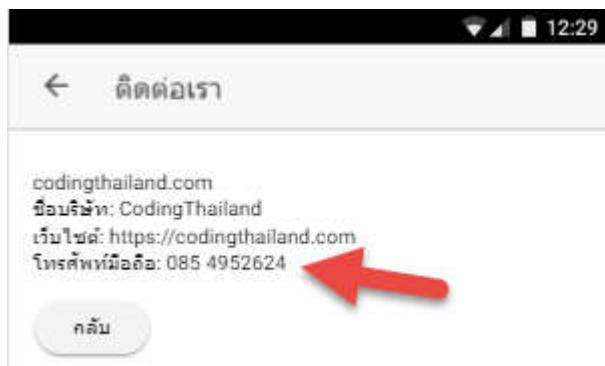
<ion-content padding>
<p>
  codingthailand.com
<br>
  ชื่อบริษัท: {{companyName}} <br>
  เว็บไซต์: {{companyWebsite}} <br>
  โทรศัพท์มือถือ: {{ phone }}
</p>
<button ion-button color="light" round navPop>กลับ</button>
</ion-content>

```

5. ลองทดสอบ App รีเม็จจากเบ็ดหน้า about ตัว Storage จะเพิ่มข้อมูล key ชื่อว่า phone พิ้งก์ก์กำหนดค่าเรียบเบ็ดอย หากใช้ Browser สามารถตรวจสอบข้อมูลได้ ดังรูป



ต่อมาทดสอบ คลิกปุ่ม “เบ็ดหน้าเพจติดต่อเรา” จะเห็นว่าค่าเบอร์โทรศัพท์จะถูก get และแสดงผลข้อมูลเรียบร้อย ดังรูป



แต่ถ้าหากปิดหรือออกจากเบราว์เซอร์ โทรศัพท์ และ key phone ที่สร้างไว้ ก็จะถูกลบออกไปเรียบร้อย ดังนี้  
(อย่าลืมกดปุ่ม refresh หมายเลขอ 1 ในรูปก่อน)

#	Key	Value
	_ionickv	local-forage-detect-blob-

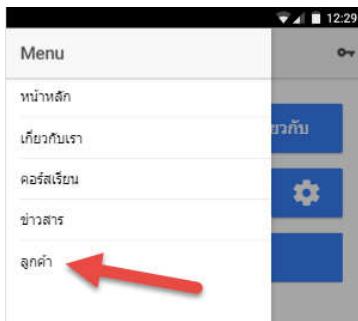
## การใช้งาน Ionic Storage เพื่อเก็บข้อมูลแบบฐานข้อมูล (SQLite)

ในหัวข้อนี้ต้องทดสอบกับเครื่องจริง หรือใช้ Ionic View หรือใช้ Ionic Dev App ก็ได้ครับ เพราะเราจะใช้ SQLite และใช้ภาษา SQL สำหรับจัดการข้อมูลต่างๆ ในรูปแบบของฐานข้อมูล เช่น การเพิ่มข้อมูล การลบข้อมูล เป็นต้น

Note: การใช้งานบนเครื่องจริง แนะนำให้ลองใช้ Ionic Dev App ดูวิธีการใช้งานได้ที่นี่ <https://youtu.be/AByQD0TNEsA>

### ขั้นตอนการใช้งานฐานข้อมูลแบบ SQL (SQLite) มีดังนี้

- ติดตั้ง sqlite ด้วยคำสั่ง ionic cordova plugin add cordova-sqlite-storage แล้วกด enter ต่อด้วยคำสั่ง npm install --save @ionic-native/sqlite แล้วกด enter จากนั้น สร้างเพจใหม่ชื่อว่า customer ด้วยคำสั่ง ionic g page customer พื้นที่จะมีไฟล์ src\app\app.component.ts ด้วย ดังรูป



- เปิดไฟล์ src\app\app.module.ts เพื่อ import SQLite และกำหนดที่ providers ดังรูป

```

1 app.module.ts ✘
2
3 ① import { SQLite } from '@ionic-native/sqlite';
4
5 ② @NgModule({
6    declarations: [
7      MyApp,
8      HomePage,
9      ListPage,
10     AboutPage,ContactPage,CoursePage,NewsPage,CourseDetailPage,
11   ],
12   imports: [
13     BrowserModule,
14     HttpModule,
15     IonicModule.forRoot(MyApp),
16     IonicStorageModule.forRoot()
17   ],
18   bootstrap: [IonicApp],
19   entryComponents: [
20     MyApp,
21     HomePage,
22     ListPage,
23     AboutPage,ContactPage,CoursePage,NewsPage,CourseDetailPage,
24   ],
25   providers: [
26     StatusBar,
27     SplashScreen,
28     ③ SQLite,
29     {provide: ErrorHandler, useClass: IonicErrorHandler},
30   ]
31 })
32 
```

3. เปิดไฟล์ src\pages\customer\customer.html เพื่อสร้างฟอร์มในรูปแบบของ template driven forms และเพิ่ม List Component ในรูปแบบของ Sliding List เพื่อให้ผู้ใช้สามารถได้รับข้อมูลได้ ให้เขียนโค้ด ดังนี้

```

<ion-header>
  <ion-toolbar>
    <button ion-button icon-only menuToggle>
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-title>รายชื่อลูกค้า</ion-title>
  <ion-buttons end>
    <button ion-button icon-only color="royal" (click)="openForm()">
      <ion-icon name="person-add"></ion-icon>
    </button>
  </ion-buttons>
</ion-toolbar>
</ion-header>

<ion-content padding>
  <form *ngIf="isToogle" novalidate #myForm="ngForm" (ngSubmit)="add(myForm.value)">
    <ion-list>
      <ion-item>
        <ion-label floating>ชื่อ-สกุล</ion-label>
        <ion-input #fullname="ngModel" ngModel type="text" name="fullname" required></ion-input>
      </ion-item>

      <ion-item>
        <ion-label floating>เบอร์โทรศัพท์</ion-label>
        <ion-input #phone="ngModel" ngModel type="text" name="phone" required></ion-input>
      </ion-item>

    <div padding>
      <button ion-button block type="submit" [disabled]="!myForm.form.valid">บันทึกข้อมูล</button>
    </div>
  </form>
</ion-content>

```

```

</ion-list>
</form>

<ion-list>
  <ion-item-sliding *ngFor="let item of customers">
    <ion-item>
      <h3>{{item.fullname}}</h3>
      <p>{{item.phone}}</p>
    </ion-item>
    <ion-item-options side="right">
      <button ion-button color="danger" (click)="delete(item)">
        <ion-icon name="trash"></ion-icon>
        ลบ
      </button>
    </ion-item-options>
  </ion-item-sliding>
</ion-list>
</ion-content>

```

4. เปิดไฟล์ src\pages\customer\customer.ts ให้เขียนโค้ดเพื่อ import SQLite เข้ามาใช้งาน รวมถึงการเปิดฐานข้อมูล การสร้างตาราง การเพิ่มข้อมูล และลบข้อมูล ดังนี้

```

import { Component } from '@angular/core';
import { NavController, NavParams, Platform, ToastController,
AlertController} from 'ionic-angular';
import { SQLite, SQLiteObject } from '@ionic-native/sqlite';

@Component({
  selector: 'page-customer',
  templateUrl: 'customer.html'
  //providers: [SQLite]
})
export class CustomerPage {
  customers: Array<Object>;
  isToggle:boolean = false;

  constructor(public sqlite:SQLite,
             public toastCtrl: ToastController,

```

```

        public platform: Platform,
        public alertCtrl: AlertController,
        public navCtrl: NavController,
        public navParams: NavParams) {
    this.platform.ready().then(() => { //ถ้า platform พร้อมใช้งาน
        this.sqlite.create({ //สร้างฐานข้อมูลชื่อว่า data.db
            name: "data.db",
            location: "default"
        }).then((db:SQLiteObject) => {
            //หากยังไม่มีตาราง customer ก็ให้สร้างตารางใหม่ หากมีแล้วก็ไม่ต้องสร้าง
            db.execSQL("CREATE TABLE IF NOT EXISTS customer (id
INTEGER PRIMARY KEY AUTOINCREMENT, fullname TEXT, phone TEXT)",
            {}).then((data) => {
                console.log("TABLE CREATED: ", data);
            }, (error) => {
                //console.error("ไม่สามารถรันคำสั่ง sql ได้", error);
            });
            //this.add();
        }, (error) => {
            //console.error("ไม่สามารถเปิดฐานข้อมูลได้", error);
        });
    });
}
//เป็น method ที่รับค่าข้อมูลจากฟอร์ม พร้อมทั้งเพิ่มข้อมูลลงในตาราง customer หาก
//เรียบร้อยก็ให้แสดง Toast Controller ว่าเพิ่มข้อมูลเรียบร้อย
public add(form) {
    this.sqlite.create({
        name: "data.db",
        location: "default"
    }).then((db:SQLiteObject) => {
        db.execSQL("INSERT INTO customer (fullname, phone) VALUES
(?,?)", [form.fullname,form.phone]).then((data) => {
            let toast = this.toastCtrl.create({
                message: 'เพิ่มข้อมูลเรียบร้อย',
                duration: 3000
            });
            toast.present();
            this.showData();
            this.isToogle = false;
        }, (error) => {
            //console.log("error: " + JSON.stringify(error));
        });
    });
}
//เป็น method ที่เขียนไว้สำหรับแสดงข้อมูลทั้งหมดจากตาราง customer โดยเรียงลำดับ
//ตามคอลัมน์ id จากมากไปหาน้อย พร้อมทั้งวนลูปเก็บข้อมูลให้กับตัวแปร customers
public showData() {
    this.sqlite.create({

```

```

        name: "data.db",
        location: "default"
    }).then((db:SQLiteObject) => {
        db.executeSql("SELECT * FROM customer ORDER BY id DESC",
        []).then((data) => {
            this.customers = [];
            if(data.rows.length > 0) {
                for(var i = 0; i < data.rows.length; i++) {
                    this.customers.push({id:
data.rows.item(i).id, fullname: data.rows.item(i).fullname, phone:
data.rows.item(i).phone});
                }
            },
            (error) => {
// console.log("error: " + JSON.stringify(error));
            });
        });
    });

// เป็น method ที่รับค่า item และรับรหัสลูกค้าเพื่อทำการลบข้อมูลในรายการนั้นๆ
public delete(item) {
    let confirm = this.alertCtrl.create({
    title: 'ยืนยันการลบข้อมูล',
    message: ``,
    buttons: [
        {
            text: 'ยกเลิก',
            handler: () => {
// console.log('cancel');
            } },
        {
            text: 'ตกลง',
            handler: () => {
                this.sqlite.create({
                    name: "data.db",
                    location: "default"
                }).then((db:SQLiteObject) => {
                    db.executeSql('DELETE FROM customer WHERE id=?',
[item.id]).then((data) => {
                        let toast = this.toastCtrl.create({
                            message: 'ลบข้อมูลเรียบร้อย',
                            duration: 3000
                        });
                        toast.present();
                        this.showData();
                    }, (error) => {
//console.log("error: " +
JSON.stringify(error));
                });
            });
        }
    ];
}

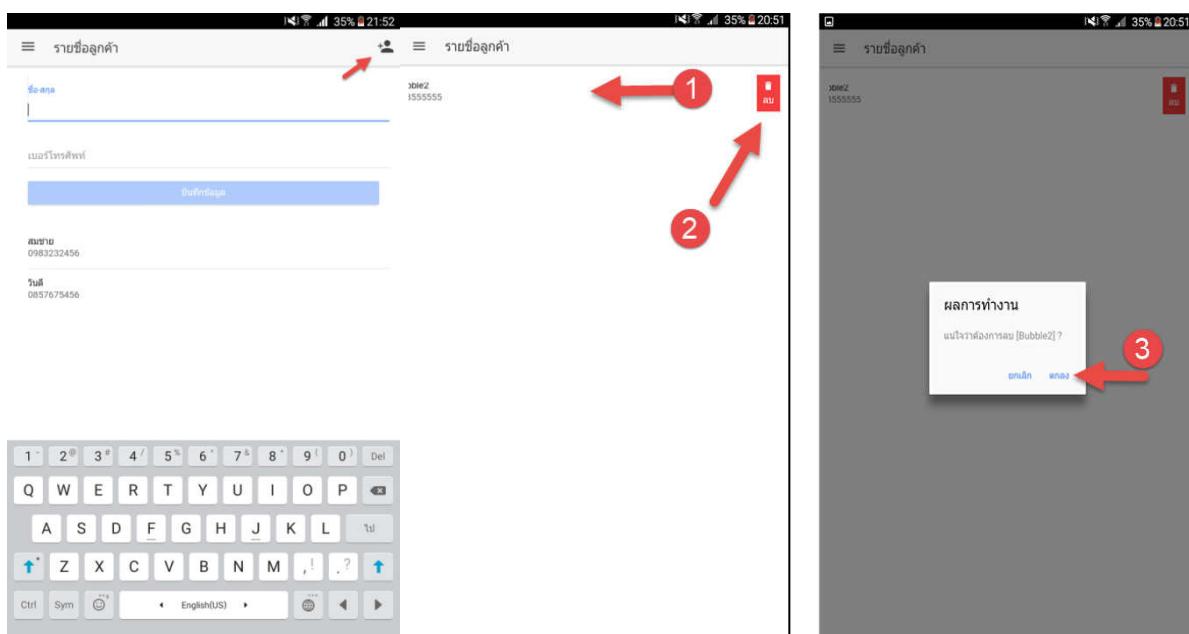
```

```

        });
    });
}
} ] });
confirm.present();
}
ionViewWillEnter() {
    this.showData();
}
//เป็น method ที่มีไว้ซ่อนหรือแสดงฟอร์ม
openForm() {
    this.isToogle = !this.isToogle;
}
}
}

```

5. ทดสอบรัน App โดยต้อง รันบนเครื่องจริง (USB), Emulator หรือ Ionic View เท่านั้น



Note: คุณสามารถเขียน sqlite ดูเพิ่มเติมได้ที่ <https://ionicframework.com/docs/native/sqlite/>

## บทที่ 10 การเขียน Ionic เชื่อมต่อกับ Firebase

Firebase เป็นบริการของ google ที่เปิดให้เราสามารถพัฒนา Web หรือ Mobile App ได้สะดวกมากขึ้น โดย firebase จะมีบริการต่างๆ คือสนับสนุนนักพัฒนา ทั้งในส่วนของการทำงานกับ Backend และอื่นๆ ทำให้ประหยัดเวลาไปได้มาก นอกจากนี้ตัว firebase ยังมี เวอร์ชันพรีให้ใช้อีกด้วย

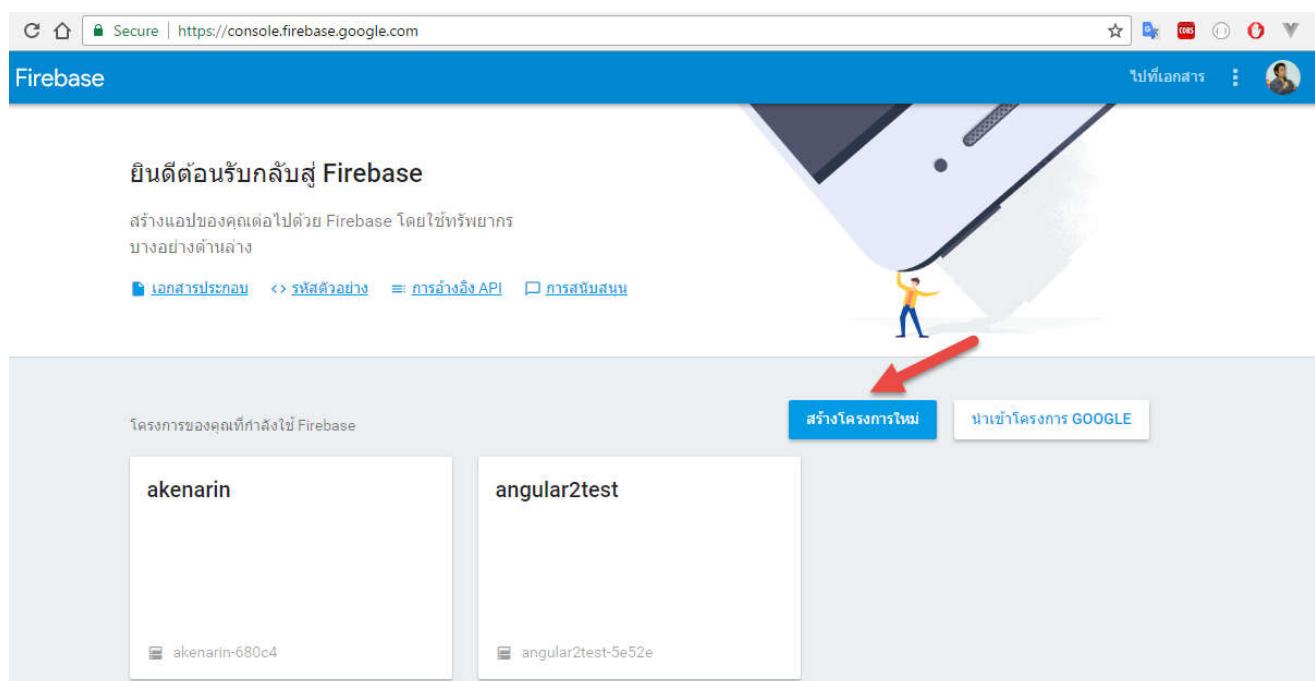
### คุณสมบัติของ Firebase มีดังนี้

- Realtime Database เป็นฐานข้อมูลแบบ NoSQL และรองรับ Data แบบ Real time
- User Authentication สำหรับไว้จัดการผู้ใช้และระบบล็อกอิน
- Static Hosting สำหรับเว็บ App
- Cloud Messaging เป็นระบบ Messaging สำหรับทำ Push Notification
- Storage (Store files) เป็นพื้นที่คุณให้บริการเก็บไฟล์ต่างๆ (ฝากไฟล์)

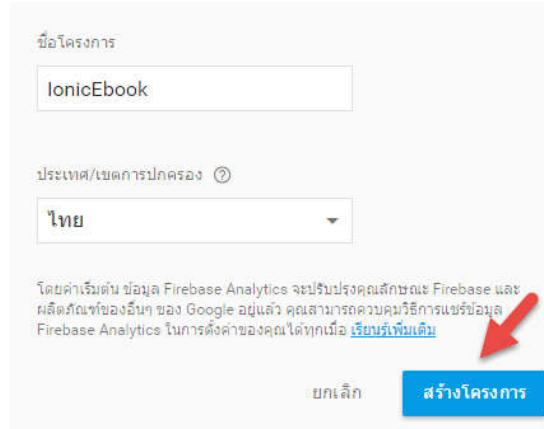
เว็บไซต์หลักของ Firebase คือ <https://firebase.google.com/> แนะนำให้สมัครสมาชิก และล็อกอินเข้าใช้งานเลย จากนั้นให้ไปที่ Console ตามลิงก์ <https://console.firebaseio.com/>

ในบทนี้เราจะมาลองใช้ Realtime Database กัน ขั้นตอนการใช้งาน มีดังนี้

1. ให้คลิกที่ปุ่ม “สร้างโครงการใหม่”



2. กรอกรายละเอียดของชื่อโครงการ และประเทศ คลิกปุ่ม “สร้างโครงการ” ดังรูป



3. ใน Workshop นี้ เราจะมาสร้างกฎความปลอดภัยง่ายๆ เพื่อเป็นกรณีศึกษา ก่อน คือ ทำให้ฐานข้อมูลของเราสามารถอ่าน และเขียนข้อมูลได้ (เป็น public) ให้ทำงานขั้นตอน ดังรูป แต่หากใช้งานจริงแนะนำให้ตั้งกฎความปลอดภัยใหม่

```

{
  "rules": {
    ".read": true,
    ".write": true
  }
}

```

Note: ดูเรื่องกฎความปลอดภัยของ firebase ได้ที่นี่ <https://firebase.google.com/docs/database/security/quickstart>

4. หากคลิกที่ Tab ข้อมูลจะเห็นว่าตอนนี้ฐานข้อมูลของเราไม่มีข้อมูลอะไร (null)

ionicebook-fd8c4:	null	+	x
-------------------	------	---	---

5. การเขียน Ionic เรื่องต่อ กับ firebase แนะนำให้ใช้ไลบรารี ชื่อว่า AngularFire2 เราต้องติดตั้งก่อน โดยใช้คำสั่ง  
 npm install firebase@4.10.1 angularfire2@5.0.0-rc.6 --save และกด enter

Note: รายละเอียดของ AngularFire2 ดูได้ที่นี่ <https://github.com/angular/angularfire2>

6. ติดตั้ง polyfill ด้วยคำสั่ง **npm install promise-polyfill --save**

7. กลับมาที่เว็บไซต์ Firebase คลิกที่เมนู Overview และคลิกเลือกที่ “เพิ่ม Firebase ไปยังเว็บแอปของคุณ”



8. จะปรากฏหน้าต่างสำหรับแสดงค่า config ต่างๆ เช่น apiKey, databaseURL เป็นต้น ซึ่งเราจะต้อง copy ไปวางในข้อ 8

```

<script src="https://www.gstatic.com/firebasejs/3.6.10.firebaseio.js"></script>
<script>
// Initialize Firebase
var config = {
  apiKey: "AIzaSyCt1EQ4DD6ui-9wM5j2dGtiqZcSP3c7R1A",
  authDomain: "ionicebook-fd8c4.firebaseio.com",
  databaseURL: "https://ionicebook-fd8c4.firebaseio.com",
  storageBucket: "ionicebook-fd8c4.appspot.com",
  messagingSenderId: "837217879658"
};
firebase.initializeApp(config);
</script>

```

ดูแหล่งข้อมูลต่อไปนี้เพื่อเรียนรู้  
[เพิ่มเติมเกี่ยวกับ Firebase](#)  
 สำหรับเว็บแอป:

[Get Started with Firebase for Web Apps](#) [Firebase Web SDK API Reference](#) [Firebase Web Samples](#)

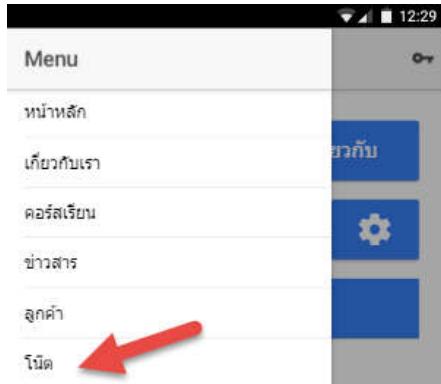
9. เปิดไฟล์ src\app\app.module.ts เพื่อ import AngularFireModule ,AngularFireDatabaseModule และ AngularFireDatabase จากนั้นตั้งค่า config (คัดลอกค่า จากข้อ 7) และ import Module ทั้งสองตัว เพื่อ init และใช้งาน และกำหนด AngularFireDatabase ที่ส่วนของ providers ด้วย ดังรูป

```

A app.module.ts x
21 import { AngularFireModule } from 'angularfire2';
22 import { AngularFireDatabaseModule, AngularFireDatabase } from 'angularfire2/database';
23
24 export const config = {
25   apiKey: "AIzaSyCt1EQ4DD6ui-9wM5j2dGtiqZcSP3c7RlA",
26   authDomain: "ionicebook-fd8c4.firebaseio.com",
27   databaseURL: "https://ionicebook-fd8c4.firebaseio.com",
28   projectId: "ionicebook-fd8c4",
29   storageBucket: "ionicebook-fd8c4.appspot.com",
30   messagingSenderId: "837217879658"
31 };
32
33 @NgModule({
34   declarations: [
35     MyApp,
36     HomePage,
37     ListPage
38   ],
39   imports: [
40     BrowserModule,
41     HttpClientModule,
42     IonicModule.forRoot(MyApp),
43     IonicStorageModule.forRoot(),
44     AngularFireModule.initializeApp(config),
45     AngularFireDatabaseModule,
46     AboutPageModule, ContactPageModule,
47     CoursePageModule, NewsPageModule, CourseDetailPageModule,
48     YoutubePageModule, SignupPageModule, CustomerPageModule
49   ],
50
51   providers: [
52     StatusBar,
53     SplashScreen,
54     SQLite,
55     {provide: ErrorHandler, useClass: IonicErrorHandler},
56     CourseServiceProvider,
57     NewsServiceProvider,
58     AuthServiceProvider,
59     AngularFireDatabase
60   ]
61 })
62 export class AppModule {}

```

10. สร้างเพจใหม่ ชื่อว่า note โดยใช้คำสั่ง ionic g page note และกด enter จากนั้นให้ import และสร้างเมนูด้านซ้ายด้วยตัวเอง



11. เปิดไฟล์ src\pages\note\note.html เพื่อสร้างฟอร์มสำหรับจัดการข้อมูล ทั้งเพิ่มและแก้ไข (ในหัวข้อนี้เราจะเขียนฟอร์มและผูกข้อมูล input ในรูปแบบของ two-way binding) จากนั้นให้ List Component แสดงข้อมูล เพิ่มโค้ด ดังนี้

```

<ion-header>
  <ion-toolbar>
    <button ion-button icon-only menuToggle>
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-title>โน๊ต เตือนความจำ</ion-title>
    <ion-buttons end>
      <button ion-button icon-only color="royal" (click)="openForm()">
        <ion-icon name="add-circle"></ion-icon>
      </button>
    </ion-buttons>
  </ion-toolbar>
</ion-header>

<ion-content padding>

  <form *ngIf="isToogle" novalidate #note="ngForm"
  (ngSubmit)="save(note.value)">
    <ion-list>
      <ion-item>
        <ion-label floating>หัวข้อ</ion-label>
        <ion-input [(ngModel)]="topic" type="text" name="topic"
        required></ion-input>
      </ion-item>

      <ion-item>
        <ion-label floating>วันที่ต้องทำ</ion-label>
        <ion-datetime [(ngModel)]="dueDate" type="text"
        displayFormat="MM/DD/YYYY" name="dueDate" required></ion-datetime>
      </ion-item>
    </ion-list>
  </form>
</ion-content>

```

```

        <div padding>
            <button ion-button type="submit" [disabled]="!note.form.valid">ເພີ່ມ
</button>
            <button ion-button type="button" [disabled]="!note.form.valid"
(click)="update(note.value)">ແກ້ໄຂ</button>
        </div>

    </ion-list>
</form>

<ion-list>
    <ion-item-sliding *ngFor="let item of items | async"
(click)="select(item)">
    <ion-item>
        <h3>{{item.note.topic}}</h3>
        <p>{{item.note.dueDate}}</p>
    </ion-item>
    <ion-item-options side="right">
        <button ion-button color="danger" (click)="delete(item)">
        <ion-icon name="trash"></ion-icon>
        ລບຂໍ້ມູນ
        </button>
    </ion-item-options>
</ion-item-sliding>
</ion-list>

</ion-content>

```

12. ເປີດໄຟລ໌ src\pages\note\note.ts ເພື່ອເຂົ້າແນວໂຄ້ດຈັດກາງກັບສ້າງຂໍ້ມູນຈາກ firebase ດັ່ງນີ້

```

import { Component } from '@angular/core';
import { NavController, NavParams } from 'ionic-angular';
import { AngularFireDatabase, AngularFireList } from
'angularfire2/database';

import { Observable } from 'rxjs/Observable';
import 'rxjs/add/operator/map';

@Component({
  selector: 'page-note',
  templateUrl: 'note.html'
})
export class NotePage {
  itemsRef: AngularFireList<any>;
  items: Observable<any[]>;

```

```

key:string;
topic:string;
dueDate:string;
isToogle:boolean = false;

constructor(private af: AngularFireDatabase,
            public navCtrl: NavController,
            public navParams: NavParams) {}

ionViewWillEnter() {
  this.showData();
}

//แสดงข้อมูลทั้งหมดจากฐานข้อมูล
showData() {
  this.itemsRef = this.af.list('/');
  // Use snapshotChanges().map() to store the key
  this.items = this.itemsRef.snapshotChanges().map(changes => {
    return changes.map(c => ({ key: c.payload.key, ...c.payload.val() }));
  });
}

//เลือก item พร้อมทั้งกำหนดค่าให้กับ input ฟอร์ม และเก็บ key ไว้ด้วยสำหรับการแก้ไข
select(item) {
  //console.log(item);
  this.topic = item.note.topic;
  this.dueDate = item.note.dueDate;
  this.key = item.key;
}

//บันทึกข้อมูล
save(note: any) {
  //console.log(blog);
  this.itemsRef.push({note});
  this.isToogle = false;
}

//อัปเดตข้อมูลตาม key ที่ส่งมา
update(note: any) {
  if (this.key) {
    this.itemsRef.update(this.key, {note});
    this.isToogle = false;
  }
}

//ลบข้อมูลแต่ละรายการตาม key ที่เลือก
delete(item:any) {
  this.itemsRef.remove(item.key);
}

```

```

        this.isToogle = false;
    }

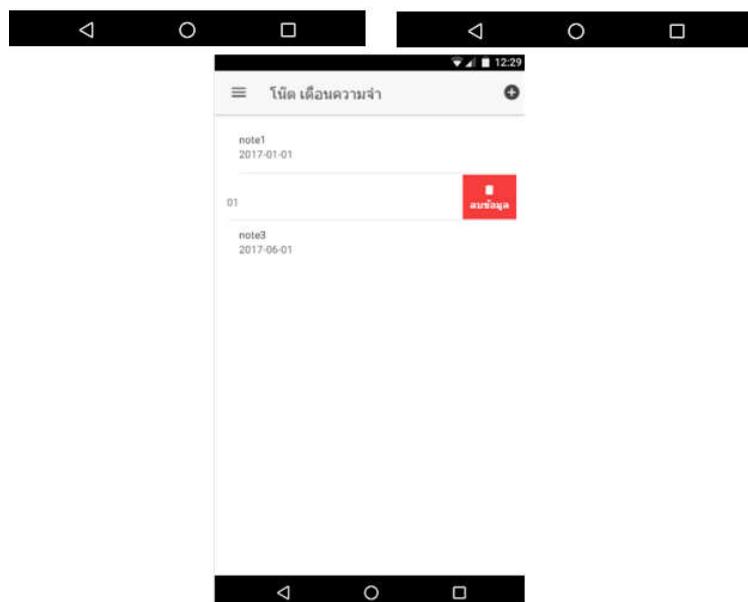
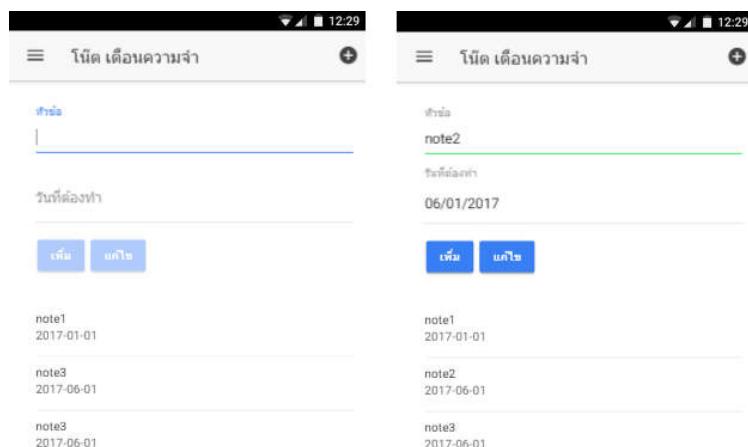
/*deleteAll() { //ลบทั้งหมด
    this.items.remove();
    this.isToogle = false;
}*/



//เป็น method ที่มีไว้ซ่อนหรือแสดงฟอร์ม
openForm() {
    this.isToogle = !this.isToogle;
}
}
}

```

13. บันทึกไฟล์ทั้งหมด แล้วทดสอบรัน App อีกครั้ง จากนั้นลองใช้งานดู



## บทที่ 11 การใช้งาน Ionic Native

Ionic Native ช่วยให้เราเรียกใช้งานความสามารถต่างๆ ของเครื่องจริง (native) ยกตัวอย่างเช่น กล้อง, geolocation, contacts เป็นต้น การใช้งานเพียงแค่ติดตั้ง plugin ลงใน App จากนั้นก็ศึกษาได้ และการใช้งานของ native แต่ละตัวก็เป็นอันเรียบง่าย (ง่ายมาก)

Ionic Native ถูกเขียนขึ้นด้วย ES5/ES6/TypeScript ซึ่งครอบคลุม Cordova/PhoneGap plugins จึงทำให้การเขียน native เป็นเรื่องง่ายจริงๆ (แน่นอนว่าคนเขียนไม่ได้แล้ว) ในบทนี้จะยกตัวอย่าง native 2 ตัวด้วยกัน ได้แก่ Camera และ Barcode Scanner หากเราต้องการใช้งานแล้วก็สามารถประยุกต์ใช้กับ Ionic Native ตัวอื่นๆ ได้เช่นเดียวกัน

**Note:** รายการ Ionic Native ทั้งหมด ดูรายละเอียดได้ที่นี่ <http://ionicframework.com/docs/native/>

**Note:** ก่อนติดตั้ง plugin ของ ionic native อย่าลืมเพิ่ม platform ก่อน หากต้องการทดสอบบนเครื่องจริง เช่น ionic cordova platform add android เป็นต้น และการทดสอบ App ในบทนี้จะต้องรันบนเครื่องจริง, Emulator หรือ Ionic View เท่านั้นถึงจะเห็นผลลัพธ์

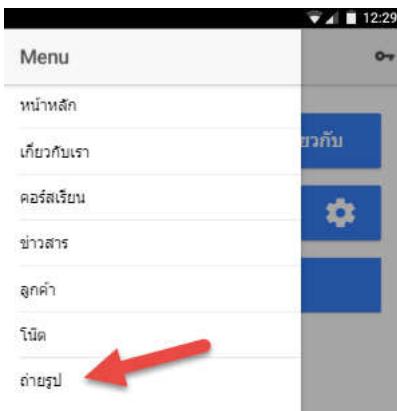
### การใช้งานกล้อง (Camera)

ขั้นตอนการใช้งาน Camera มีดังนี้

- ติดตั้ง plugin Camera ลงใน App ของเราก่อน ด้วยคำสั่ง (อย่าลืมกด enter)

```
ionic cordova plugin add cordova-plugin-camera
npm install --save @ionic-native/camera
```

- สร้างเพจใหม่ขึ้นมา ด้วยคำสั่ง ionic g page camera จากนั้นอย่าลืม import และทำเมนูขึ้นด้วย ดังรูป



3. เปิดไฟล์ src\app\app.module.ts เพื่อ import Camera เข้ามาใช้งาน และต้องกำหนดในส่วนของ providers ด้วย ดังรูป

```

43 | import { Camera } from '@ionic-native/camera'; ← 1
44 |
45 | @NgModule({
46 |   declarations: [
47 |     MyApp,
48 |     HomePage,
49 |     ListPage
50 |   ],
51 |   imports: [
52 |     BrowserModule,
53 |     HttpClientModule,
54 |     IonicModule.forRoot(MyApp),
55 |     IonicStorageModule.forRoot(),
56 |     AngularFireModule.initializeApp(config),
57 |     AngularFireDatabaseModule,
58 |     AboutPageModule, ContactPageModule,
59 |     CoursePageModule, NewsPageModule, CourseDetailPageModule,
60 |     YoutubePageModule, SignupPageModule, CustomerPageModule, NotePageModule,
61 |     CameraPageModule
62 |   ],
63 |   bootstrap: [IonicApp],
64 |   entryComponents: [
65 |     MyApp,
66 |     HomePage,
67 |     ListPage
68 |   ],
69 |   providers: [
70 |     StatusBar,
71 |     SplashScreen,
72 |     SQLite,
73 |     {provide: ErrorHandler, useClass: IonicErrorHandler},
74 |     CourseServiceProvider,
75 |     NewsServiceProvider,
76 |     AuthServiceProvider,
77 |     AngularFireDatabase,
78 |     Camera ← 2
79 |   ]
80 | })
81 | export class AppModule {}

```

Note: การใช้งาน Ionic native ทุกตัวต้องทำขั้นตอนที่ 3 นี้เหมือนๆกัน

4. เปิดไฟล์ src\pages\camera\camera.ts เพื่อเขียน method เกี่ยวกับการถ่ายรูป และใส่ค่า options ต่างๆ ดังนี้

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';

//import Camera และ CameraOptions เข้ามาใช้งาน
import { Camera, CameraOptions } from '@ionic-native/camera';

@IonicPage()
@Component({
  selector: 'page-camera',
  templateUrl: 'camera.html'
})
export class CameraPage {
  myImg:any;

  constructor(private camera: Camera, public navCtrl: NavController,
  public navParams: NavParams) {}

  //เป็น method สำหรับถ่ายภาพ พร้อมทั้งกำหนด options ต่างๆให้กับกล้องด้วย
  takePicture() {
    //กำหนด options ให้กับ Camera
    //รายละเอียด options เพิ่มเติมได้ที่ https://github.com/apache/cordova-plugin-camera#cameraoptions-errata
    let options:CameraOptions = {
      destinationType: this.camera.DestinationType.DATA_URL,
      //เลือก sourceType เป็นกล้อง หรือหากต้องการเลือกจาก Gallery ให้ sourceType:
      Camera.PictureSourceType.SAVEDPHOTOALBUM
      sourceType: this.camera.PictureSourceType.CAMERA,
      encodingType: this.camera.EncodingType.JPEG, //เลือกนามสกุลรูปเป็น jpeg
      targetWidth: 600,
      targetHeight: 600,
      saveToPhotoAlbum: false,
      correctOrientation: true
    }
    this.camera.getPicture(options).then((imageData) => {
      // imageData is either a base64 encoded string or a file URI
      // If it's base64:
      // เมื่อถ่ายรูปสำเร็จจะ encode เป็น base64 แล้วกำหนดให้กับ myImg แล้วนำไปแสดงผลที่ template
      // ในจังหวะนี้เราจะสามารถใช้ตัวแปร this.myImg บันทึกเก็บลงฐานข้อมูลได้เลย และให้คอลัมน์ในตารางฐานข้อมูลมีชนิดเป็น
      TEXT
      //ถ้าเก็บลงฐานข้อมูลแล้ว เราจะสามารถดึงคอลัมน์นี้มาแสดงผลรูปภาพได้ทันที
      this.myImg = 'data:image/jpeg;base64,' + imageData;
    }, (err) => {
      // Handle error
    });
  }
}

```

5. เปิดไฟล์ src\pages\camera\camera.html เพื่อเขียนโค้ดปุ่มถ่ายรูป และผูกข้อมูลภาพถ่ายที่ได้ จากนั้นนำมาแสดงผล ดังนี้

```

<ion-header>
  <ion-navbar>
    <button ion-button menuToggle>
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-title>ถ่ายรูป</ion-title>
  </ion-navbar>
</ion-header>

<ion-content padding>
  <button ion-button full color="primary" (click)="takePicture()">ถ่ายรูป!!
</button>

  <ion-card *ngIf="myImg">
    <img [src]="myImg">
  </ion-card>

</ion-content>

```

6. ทดสอบวัน App บนเครื่องจริง กดปุ่มถ่ายรูป จะได้ผลลัพธ์ ดังนี้



Note: ดูรายละเอียดของ Camera ได้ที่ <https://ionicframework.com/docs/native/camera/>

## การใช้งาน Barcode Scanner และ QR CODE

ในหัวข้อนี้เราจะมาใช้ Barcode Scanner ซึ่งเป็น plugin อีกตัวหนึ่งที่น่าสนใจมาก นอกจากระ scan บาร์โค้ดทั่วไปได้แล้ว ยังสามารถ scan QR CODE ได้ด้วย การทำงานของ plugin นี้หลังจากเรียกใช้งานจะมีการเปิดกล้องของมือถือ และ scan บาร์โค้ดให้เราอัตโนมัติ างานนี้จะคืนค่าผลลัพธ์กลับคืนมาให้เราใช้งาน (อาจเป็นเลข barcode หรือ data อื่นๆ)

Note: รายละเอียดของ Barcode Scanner ได้ที่นี่ <https://ionicframework.com/docs/native/barcode-scanner/>

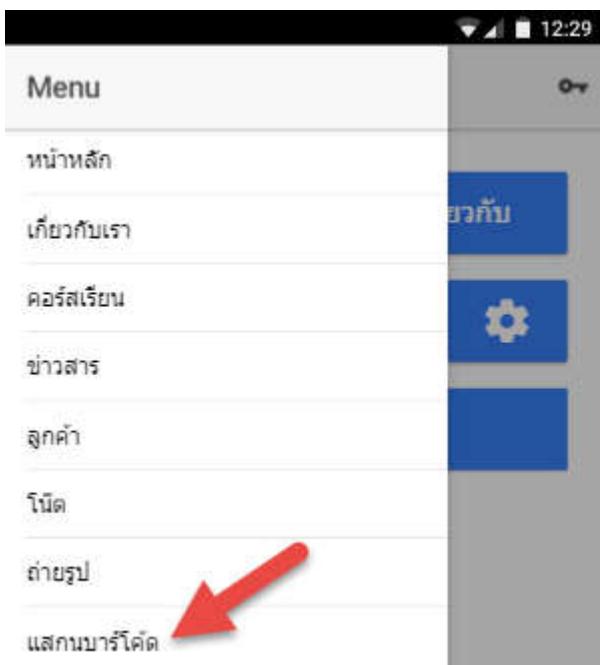
### ขั้นตอนการใช้งาน มีดังนี้

- ติดตั้ง plugin Barcode Scanner ลงใน App ของเรา ก่อน ด้วยคำสั่ง (อย่าลืมกด enter)

```
ionic cordova plugin add phonegap-plugin-barcodescanner --save
```

```
npm install --save @ionic-native/barcode-scanner
```

- สร้างเพจใหม่ขึ้นว่า camera ด้วยคำสั่ง ionic g page barcode จากนั้นอย่าลืม import และทำเมนูด้านข้างด้วย ดังรูป



3. เปิดไฟล์ src\app\app.module.ts เพื่อ import BarcodeScanner เข้ามาใช้งาน และกำหนดในส่วนของ providers

```
44 import { Camera } from '@ionic-native/camera';
45 import { BarcodeScanner } from '@ionic-native/barcode-scanner';
46
47 @NgModule({
48   declarations: [
49     MyApp,
50     HomePage,
51     ListPage
52   ],
53   imports: [
54     BrowserModule,
55     HttpClientModule,
56     IonicModule.forRoot(MyApp),
57     IonicStorageModule.forRoot(),
58     AngularFireModule.initializeApp(config),
59     AngularFireDatabaseModule,
60     AboutPageModule, ContactPageModule,
61     CoursePageModule, NewsPageModule, CourseDetailPageModule,
62     YoutubePageModule, SignupPageModule, CustomerPageModule, NotePageModule,
63     CameraPageModule, BarcodePageModule
64   ],
65   bootstrap: [IonicApp],
66   entryComponents: [
67     MyApp,
68     HomePage,
69     ListPage
70   ],
71   providers: [
72     StatusBar,
73     SplashScreen,
74     SQLite,
75     {provide: ErrorHandler, useClass: IonicErrorHandler},
76     CourseServiceProvider,
77     NewsServiceProvider,
78     AuthServiceProvider,
79     AngularFireDatabase,
80     Camera,
81     BarcodeScanner ←
82   ]
83 })
84 export class AppModule {}
```

4. เปิดไฟล์ src\pages\barcode\barcode.ts เพื่อเขียน method scan() และรับค่า data มาแสดงผล ดังนี้

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';
//import BarcodeScanner เชื่อมใช้งาน
import { BarcodeScanner } from '@ionic-native/barcode-scanner';

@IonicPage()
@Component({
  selector: 'page-barcode',
  templateUrl: 'barcode.html'
})
export class BarcodePage {

  data:any;

  constructor(private barcodeScanner: BarcodeScanner, public navCtrl: NavController, public navParams: NavParams) {}

  scan() {
    this.barcodeScanner.scan().then((barcodeData) => {
      // Success! Barcode data is here
      // เราสามารถใช้ data ที่ได้รับประยุกต์ใช้งานต่างๆได้ เช่น นำค่า data ส่งไปค้นหาในฐานข้อมูล เป็นต้น
      this.data = barcodeData.text
    }, (err) => {
      // An error occurred
    });
  }
}

```

5. เปิดไฟล์ไฟล์ src\pages\barcode\barcode.html เพื่อสร้างบูม Scan และแสดงผลข้อมูล ดังนี้

```

<ion-header>
  <ion-navbar>
    <button ion-button menuToggle>
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-title>Barcode</ion-title>
  </ion-navbar>
</ion-header>

<ion-content padding>

  <button ion-button outline color="primary" (click)="scan()">สแกนบาร์โค้ด
</button>
  <ion-card *ngIf="data">

```

```

    ข้อมูลที่ได้รับคือ {{data}}
</ion-card>

</ion-content>

```

6. ทดสอบรัน App บนเครื่องจริง กดปุ่มเพื่อ scan barcode (อย่าลืมเติยมสินค้า ที่มีبارك็อกติดกับทดสอบด้วย) จะได้ผลลัพธ์ ดังนี้



7. หากต้องการอ่าน QR CODE ให้กำหนด options เพิ่ม โดยระบุ format เป็น QR\_CODE ดังนี้ (ยกมาเฉพาะ method scan())

```

scan(): void {
    //ถ้า options ออกจะเป็น barcode
    let options = {
        formats: 'QR_CODE'
    }

    this.barcodeScanner.scan(options).then((barcodeData) => {
        // Success! Barcode data is here
        this.myBarcode = barcodeData.text;
    }, (err) => {
        // An error occurred
    });
}

```

จากนั้นก็ทดสอบอีกครั้ง เพื่อ อ่าน QR CODE

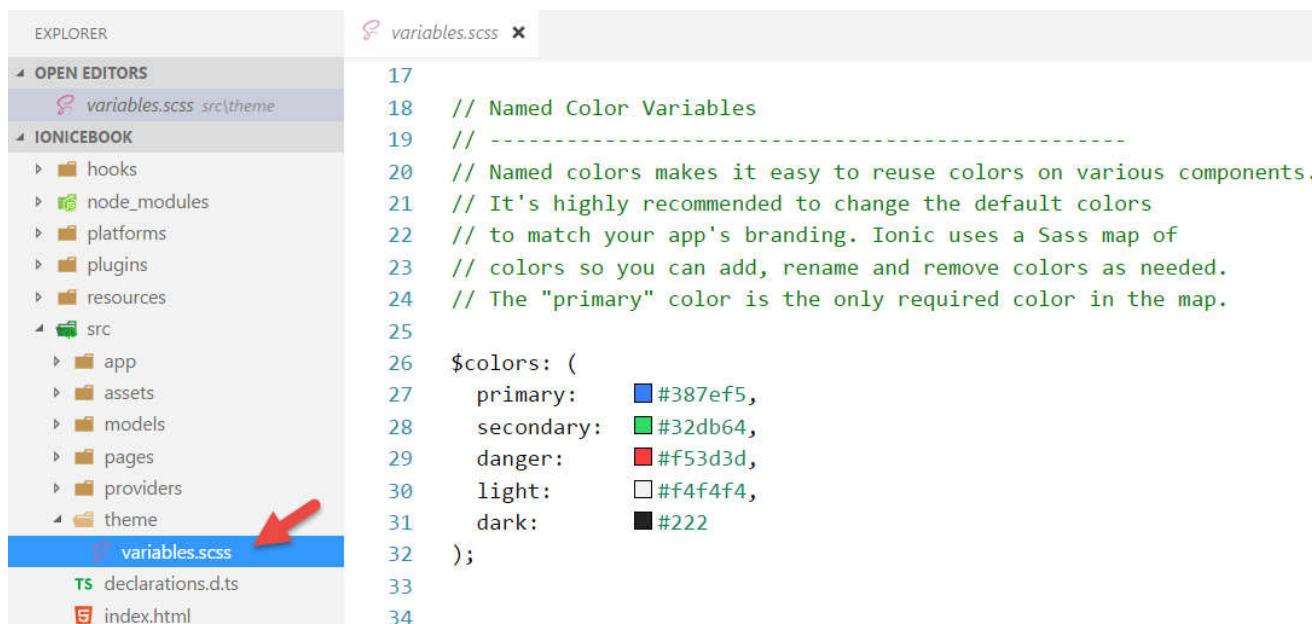


## บทที่ 12 การตกแต่ง App ด้วย Sass และ จัดรูปแบบการแสดงข้อมูลด้วย Pipes

การตกแต่งความสวยงามให้กับ App ก็คือการทำ Theme นั่นเอง ปกติแล้วถ้าจะให้ดีแนะนำศึกษาเพิ่มเติมเรื่อง Sass เพิ่มเติม แต่ถ้าเรา想 เขียน Sass ไม่กันดันนักก็สามารถเขียน CSS แบบปกติได้

### การทำ Theming ให้กับ App

ถ้าต้องการเปลี่ยนสีต่างๆ ของ Theme ให้เปิดไฟล์ src/theme/variables.scss จากนั้นสามารถแก้ไขค่าสีต่างๆ ได้เลย



```

EXPLORER variables.scss
OPEN EDITORS
IONICBOOK
hooks
node_modules
platforms
plugins
resources
src
app
assets
models
pages
providers
theme
variables.scss
declarations.d.ts
index.html

variables.scss
17
18 // Named Color Variables
19 // -----
20 // Named colors makes it easy to reuse colors on various components.
21 // It's highly recommended to change the default colors
22 // to match your app's branding. Ionic uses a Sass map of
23 // colors so you can add, rename and remove colors as needed.
24 // The "primary" color is the only required color in the map.
25
26 $colors: (
27   primary: #387ef5,
28   secondary: #32db64,
29   danger: #f53d3d,
30   light: #f4f4f4,
31   dark: #222
32 );
33
34

```

เราสามารถตั้งชื่อตัวแปรเพิ่ม และใส่ค่าสีเพิ่มได้ด้วย เช่น ในตัวอย่างนี้จะเพิ่ม mycolor เข้าไป

```
$colors: (
  primary: #387ef5,
  secondary: #32db64,
  danger: #f53d3d,
  light: #f4f4f4,
  dark: #222,
  mycolor: #008080
);
```

หากต้องการใช้ mycolor จะบุน color="ชื่อตัวแปรที่สร้างขึ้น" ให้กับ Components ที่ต้องการได้เลย เช่น

```
<button color="mycolor">I'm a button</button>
```

### Utility Attributes

Ionic ได้เตรียมตัวช่วยเพื่อให้เราเขียน Sass/CSS ได้ง่ายขึ้น โดยที่เราไม่ต้องเขียนเอง ก่อนเขียนเองแนะนำให้ดูหัวข้อนี้ก่อน เราสามารถใช้ Attributes เหล่านี้ให้กับ Components ที่เราต้องการได้เลย ประกอบด้วย

### 1. การจัดรูปแบบข้อความ ที่ใช้บ่อย

text-left	จัดข้อความซ้าย
text-center	จัดข้อความกลาง
text-right	จัดข้อความขวา
text-justify	กระจายข้อความให้พอดี
text-wrap	wrap ข้อความ
text nowrap	ไม่ wrap ข้อความ
text-uppercase	ปรับเป็นตัวพิมพ์ใหญ่
text-lowercase	ปรับเป็นตัวพิมพ์เล็ก

### 2. Element Padding ใช้สำหรับเดิมช่องว่าง หรือเว้นช่อง (ระยะห่าง) ให้กับ element

padding	เพิ่มช่องว่างให้กับ element ทั้งหมด
padding-top	เพิ่มช่องว่างด้านบน
padding-left	เพิ่มช่องว่างด้านซ้าย
padding-right	เพิ่มช่องว่างด้านขวา
padding-bottom	เพิ่มช่องว่างด้านล่าง
no-padding	ลบช่องว่างทั้งหมดออกไป

### 3. ion-buttons property จัดและเรียงปุ่ม (Button)

start	iOS จะซิดซ้าย ถ้า Android ตัวแรกจะซิดขวา
end	iOS จะซิดซ้าย ถ้า Android ตัวสุดท้ายจะซิดขวา
left	จัดปุ่มซิดด้านซ้าย
right	จัดปุ่มซิดด้านขวา

Note: ดูรายละเอียดของ Utility Attributes เพิ่มเติมได้ที่ <http://ionicframework.com/docs/theming/css-utilities/>

## การตั้งค่า App โดยการเขียนทับตัวแปร Sass ของ Ionic

เราสามารถเขียนทับตัวแปร Sass ของ Ionic ที่มีอยู่แล้วได้ การเขียนคือให้เปิดไฟล์ `src/theme/variables.scss` แล้วเขียนตัวแปรร่วมค่าที่ต้องการได้เลย ยกตัวอย่างเช่น

```
$colors: (
  primary: #387ef5,
  secondary: #32db64,
  danger: #f53d3d,
  light: #f4f4f4,
  dark: #222,
  mycolor: #008080
);
```

```
$list-background-color: #fff;
```

```
$list-ios-activated-background-color: #3aff74;
$list-md-activated-background-color: #3aff74;
$checkbox-ios-background-color-on: #32db64;
$checkbox-ios-icon-border-color-on: #fff;
```

```
$checkbox-md-icon-background-color-on: #32db64;
$checkbox-md-icon-background-color-off: #fff;
$checkbox-md-icon-border-color-off: #cecece;
$checkbox-md-icon-border-color-on: #32db64;
```

ตัวแปรที่เราสามารถเขียนทับได้ดูรายละเอียดทั้งหมดได้ที่ <https://ionicframework.com/docs/theming/overriding-ionic-variables/>

Note: หากลองแก้ไขต่างๆแล้วอย่าลืมบันทึกไฟล์แล้วลองรันดูอีกครั้ง

## จัดรูปแบบการแสดงข้อมูลด้วย Pipes

Pipes ใน Angular มีไว้สำหรับ transform, filter หรือใช้จัดรูปแบบข้อมูลต่างๆ เราสามารถแทรกเครื่องหมาย | เข้าไปใน html ได้เลย การใช้งาน Pipes ใน Ionic นั้นมี 2 วิธีคือ

1. เรียกใช้ได้เลย คือ ใน Angular จะมี Built-in pipes เราสามารถเรียกใช้ได้เลยด้วยเครื่องหมาย |
2. สร้าง custom pipe เอง และเวลาใช้งานก็ข้างชื่อ pipe นั้นๆ

### พื้นฐานการใช้งาน Built-in pipes ใน Angular

1. uppercase มีไว้แปลงตัวอักษรให้เป็นตัวพิมพ์ใหญ่ทั้งหมด

```
<p>{{ 'Coding Thailand' | uppercase }}</p>
```

<!-- ผลลัพธ์ คือ CODING THAILAND -->

2. lowercase มีไว้แปลงตัวอักษรให้เป็นตัวพิมพ์เล็กทั้งหมด

```
<p>{{ 'Coding Thailand' | lowercase }}</p>
```

<!-- ผลลัพธ์ คือ coding thailand -->

3. number (decimal) ใช้สำหรับจัดรูปแบบของตัวเลขต่างๆ เช่น

```
<p>{{ 12345 }}</p>
```

<!-- ผลลัพธ์ คือ '12345' -->

```
<p>{{ 12345 | number }}</p>
```

<!-- ผลลัพธ์ คือ '12,345' -->

```
<p>{{ 12345 | number:'6.' }}</p>
```

<!-- ผลลัพธ์ คือ '012,345' -->

<p>{{ 12345 | number:'2' }}</p>

<!-- ผลลัพธ์ คือ '12,345.00' -->

#### 4. percent จัดรูปแบบข้อมูลในรูปแบบ %

a: number = 0.259;

b: number = 1.3495;

<!--output '26%'-->

<p>A: {{a | percent}}</p>

<!--output '0,134.950%'-->

<p>B: {{b | percent:'4.3-5'}}</p>

<!--output '0 134,950 %'-->

<p>B: {{b | percent:'4.3-5':'fr'}}</p>

#### 5. currency ให้สำหรับจัดรูปแบบในรูปสกุลเงิน

a: number = 0.259;

b: number = 1.3495;

<!--output '\$0.259'-->

<p>A: {{a | currency}}</p>

<!--output 'CA\$0.26'-->

<p>A: {{a | currency:'CAD'}}</p>

<!--output 'CAD0.26'-->

```

<p>A: {{a | currency:'CAD':'code'}}</p>

<!--output 'CA$0,001.35'-->

<p>B: {{b | currency:'CAD':'symbol':'4.2-2'}}</p>

<!--output '$0,001.35'-->

<p>B: {{b | currency:'CAD':'symbol-narrow':'4.2-2'}}</p>

<!--output '0 001,35 CA$'-->

<p>B: {{b | currency:'CAD':'symbol':'4.2-2':'fr'}}</p>

```

## 6. date ใช้สำหรับจัดรูปแบบวันที่และเวลา

<p>{{ birthday | date:'ddMMyyyy' }}</p>

<!-- ผลลัพธ์ คือ '07/16/1986' -->

<p>{{ birthday | date:'longDate' }}</p>

<!-- ผลลัพธ์ คือ 'July 16, 1986' -->

<p>{{ birthday | date:'HHmmss' }}</p>

<!-- ผลลัพธ์ คือ '15:30:00' -->

<p>{{ birthday | date:'shortTime' }}</p>

<!-- ผลลัพธ์ คือ '3:30 PM' -->

Note: การใช้งาน Pipes ดูเพิ่มเติมได้ที่ <https://angular.io/docs/ts/latest/guide/pipes.html>

Note: ดูรายละเอียด Built-in pipes เพิ่มเติมได้ที่ <https://angular.io/docs/ts/latest/api/#?apiFilter=pipe>

## การสร้าง Custom Pipes ใน Ionic

สำหรับสร้างและใช้งาน Pipes ใน Ionic นั้น เราสามารถ gen code ได้โดยโดยใช้ Ionic CLI รูปแบบการใช้งานแน่นอนก็เขียนแบบ Angular รูปแบบการเรียกใช้ มีดังนี้

```
{{someData | somePipe}}
```

รูปแบบคำสั่งในการสร้าง Pipes คือ ionic g pipe <ชื่อ Pipe>

ขั้นตอนการสร้างและใช้งาน Pipes ใน Ionic มีดังนี้

1. การสร้าง Pipe ใช้คำสั่ง ionic g pipe myPipe แล้วกดปุ่ม enter

```
ionic g pipe myPipe
```

2. เปิดไฟล์ src\pipes\my\my.ts เขียนได้ดังนี้

```
import { Pipe, PipeTransform } from '@angular/core';
```

```
@Pipe({
  name: 'myUppercase', //ตั้งชื่อ pipe เวลาเรียกใช้ให้ใช้ชื่อนี้
  pure: false
})
```

```
export class MyPipe implements PipeTransform {
```

```
/*
```

การเขียน pipe จะเขียนใน method transform และให้คืนค่ากลับ  
ในตัวอย่างนี้คือการแปลงค่า (string) ที่ส่งมาไปเป็นตัวพิมพ์ใหญ่ทั้งหมด  
ถ้าหากอยากรูปแบบที่ต้องการให้เปลี่ยนได้ตรงนี้

```
*/
```

```
transform(value, args) {
```

```
  value = value + ""; // make sure it's a string
```

```

    return value.toUpperCase();
}

}

```

3. หากเพจใหม่ ต้องการเรียกใช้ Pipe ที่สร้างขึ้น ก็ให้ไป import PipesModule ที่ module ของเพจนั้นๆ เสียก่อน จึงจะสามารถใช้งานได้ ในกรณีข้อยกตัวอย่างเป็นเพจข่าวสาร (news) ให้เปิดไฟล์ src\pages\news\news.module.ts เขียนโค้ดเพื่อ import PipesModule เข้ามาใช้งาน และกำหนดที่ส่วนของ imports ดังรูป



```

news.module.ts x
1 import { NgModule } from '@angular/core';
2 import { IonicPageModule } from 'ionic-angular';
3 import { NewsPage } from './news';
4
5 import { PipesModule } from '../../../../../pipes/pipes.module'; 1
6
7 @NgModule({
8   declarations: [
9     NewsPage,
10   ],
11   imports: [
12     IonicPageModule.forChild(NewsPage),
13     PipesModule 2 ←
14   ],
15 })
16 export class NewsPageModule {}

```

4. ตอนนี้เพจข่าว (news) ก็สามารถเรียกใช้งาน Pipe นี้ได้เลย ลองเปิดไฟล์ src\pages\news\news.html เขียนโค้ด ดังนี้

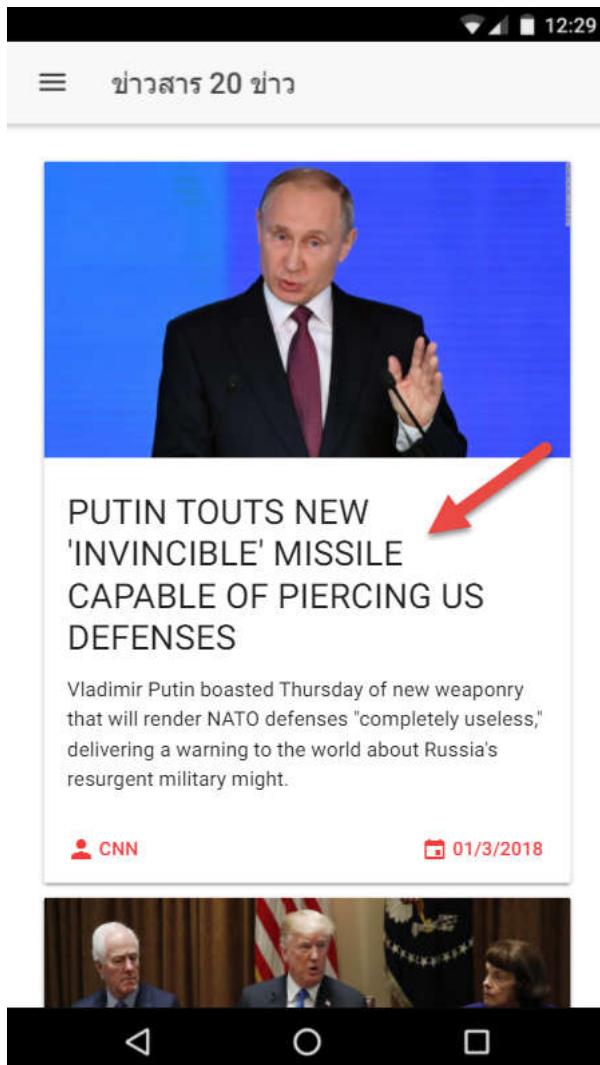


```

news.html x
6 <ion-title>ข่าวสาร {{ total }} ข่า</ion-title>
7 </ion-navbar>
8 </ion-header>
9
10 <ion-content padding>
11
12 <ion-card *ngFor="let n of articles">
13   <img *ngIf="n.urlToImage" [src]="n.urlToImage">
14   <ion-card-content>
15     <ion-card-title>
16       {{ n.title | myUppercase }} 16
17       </ion-card-title>
18     <p>
19       {{ n.description }}
20     </p>
21   </ion-card-content>
22

```

5. ບັນທຶກໄຟລ໌ ແລ້ວທົດສອບ App ອີກຄວັງ ຈະເຮັນຫວ້າຂອ້າວກລາຍເປັນຕົວພິມພົນຢູ່ທັງໝາດ



## บทที่ 13 การเตรียม Resources และ Publish App

หลังจากที่เราได้สร้าง App เรียบร้อย ขั้นตอนต่อไปคือการเตรียม App ของเราเพื่ออัปโหลดขึ้น Store มาดูกันว่าเราไม่อะไรที่ต้องทำกันบ้าง

### ตั้งค่า Bundle ID, App Name, เลขเวอร์ชัน สำหรับ App

ต่อมาก็เป็นตั้งค่า Bundle ID ซึ่งจะไม่ซ้ำกับ App อื่นๆ ในโลกนี้ หลักการตั้งเราอาจตั้งในรูปแบบเช่น com.yourdomain.yourproject ก็ได้ นอกจานั้นก็ตั้งชื่อ App พร้อมเลข version (หากมีการเปลี่ยนเวอร์ชัน และอัปโหลดขึ้น Store อีครั้งอย่าลืมเปลี่ยนตัวเลขด้วย)

เปิดไฟล์ config.xml ขึ้นมา แล้วแก้ไขรายละเอียดให้เป็นของตัวเอง

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<widget id="com.codingthailand.ionicebook" version="1.0.1" xmlns="http://www.w3.org/ns/widgets" xmlns:cdv="http://cordova.apache.org/ns/1.0">
  <name>Ionic Ebook</name>
  <description>Ionic Ebook V.2</description>
  <author email="codingthailand@gmail.com" href="https://codingthailand.com/">CodingThailand Team</author>
  <content src="index.html"/>
  <access origin="*"/>

```

### การลดขนาดรูปภาพ ก่อน Build App

เพื่อให้ App เราโหลดเร็วที่สุด สำคัญต้องคือ การลดขนาดของรูปภาพทั้งหมดใน App ของเรา แนะนำให้ทำด้วย อาจใช้เครื่องมือนี้ได้เลย <http://tinypng.com/>

หลังจากเสร็จขั้นตอนด้านบนนี้ แนะนำให้รัน App ดูใน Emulator หรือ USB อีกรอบว่าไอคอน และ splash screen ของเราหรือยัง

### การสร้างไอคอน, การ Publish App ทั้ง Android และ iOS

การสร้างไอคอน, การ Publish App ทั้ง Android และ iOS มีรายละเอียดหลายขั้นตอน ผู้เขียนจึงได้จัดทำเป็นรูปแบบวิดีโอสามารถดาวน์โหลดศึกษาได้ตามลิงก์นี้

<http://bit.ly/2LNduPD>

**Note:** ในวิดีโอนี้หากมีคำสั่งเกี่ยวกับ cordova อย่าลืมเติมคำว่า cordova เข้าไปหลังคำสั่ง ionic ด้วย เช่น ionic plugin add ก็ให้เพิ่มเป็น ionic cordova plugin add เป็นต้น

## บทที่ 14 ใบนัสพิเศษ

- ตัวอย่างไฟล์ psd (PhotoShop) สำหรับทำ splash screen

ดาวน์โหลดได้ที่นี่ <http://bit.ly/2M4HIdQ>

- ไฟล์โค้ดทั้งหมดในหนังสือ

ดาวน์โหลดได้ที่นี่ <http://bit.ly/2LRNfaE>

ยินดีด้วยครับ! คุณได้พัฒนาตัวเองอีกขั้นแล้ว  
หวังว่าหนังสือเล่มนี้จะช่วยพัฒนาความรู้ และชีวิตทุกคนให้ดีขึ้นนะครับ  
หากมีข้อสงสัยตรงไหนก็สอบถามมาได้ทางแฟนเพจครับ

ขอบคุณมากครับ

โค้ชเอก

Codingthailand.com