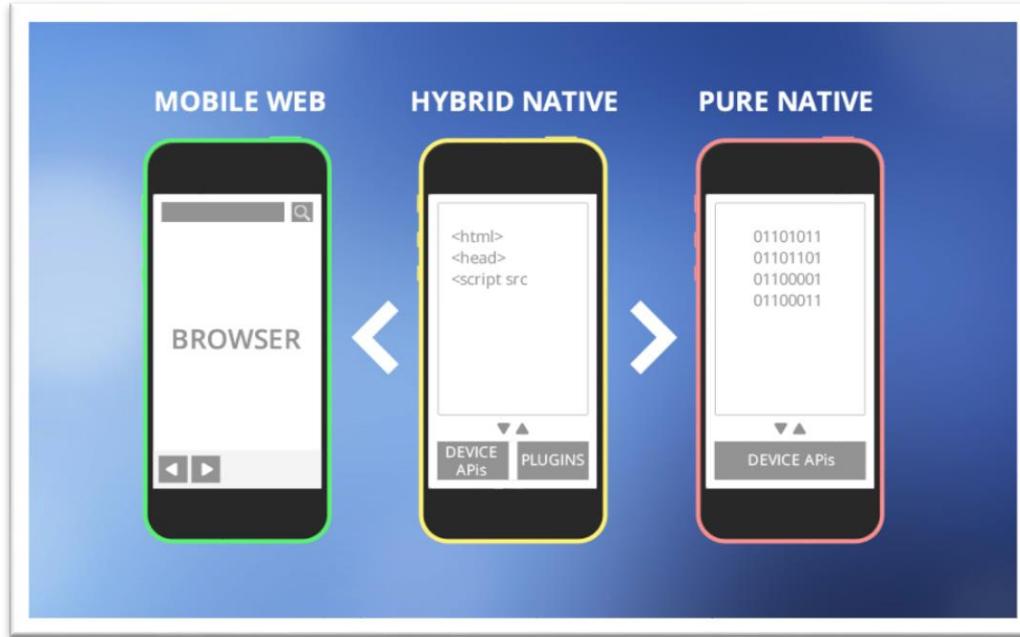


Module 1: รู้จักกับการพัฒนา Mobile Application แบบ Hybrid



แนวทางการพัฒนาแอพพลิเคชัน สำหรับการทำงานบนอุปกรณ์พกพา (mobile device) ทำได้ 3 รูปแบบ หลักๆ ด้วยกันคือ

แนวทางที่ 1 การสร้าง mobile web ซึ่งผู้ใช้งานสามารถเข้าใช้งานผ่านเว็บ Browser โดยมีการออกแบบเว็บไซต์ให้รองรับการแสดงผลบนอุปกรณ์พกพา ด้วยเทคนิคต่าง ๆ เช่น Responsive web design เป็นต้น

แนวทางที่ 2 การสร้างแอพแบบ Hybrid native เทคนิคนี้เป็นการผสมผสานระหว่าง การนำเทคโนโลยีเว็บไซต์ (html ,css,js) มาทำงานร่วมกับภาษา native (java, objective c, swift, kotlin) โดยมีเครื่องมือจาก third party มาช่วยในการแปลงเว็บไซต์ให้เป็นแอพแบบ native ได้หลายหลายแพลตฟอร์ม โดยเครื่องมือที่ได้รับความนิยม เช่น Phonegap/Cordova และ ionic framework เป็นต้น

แนวทางที่ 3 การสร้างแอพแบบ Pure Native คือการพัฒนาแอพโดยใช้ภาษาของระบบนั้น ๆ เลย ๆ เช่น Android ใช้ Java และ Kotlin ส่วน iOS ใช้ Objective C และ Swift เป็นภาษาหลักในการพัฒนาแอพพลิเคชัน

บทสรุป

Native app

พัฒนาด้วยภาษาโปรแกรมที่ถูกกำหนดมาใจแต่ละ platform เลย เช่น

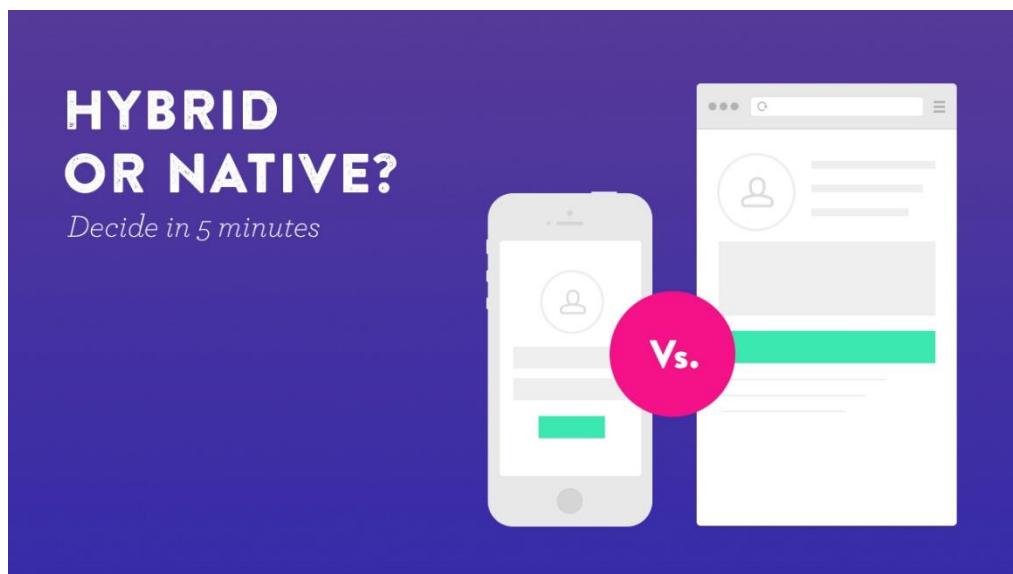
- Android app พัฒนาด้วย Java, Kotlin และ C++ เป็นต้น
- iOS app พัฒนาด้วย Objective-C และ Swift เป็นต้น

Hybrid app

เป็น app ที่จะมี Web View หรือส่วนการแสดงผล web ซึ่งจะอยู่ใน Native app ทำให้สามารถเราสามารถ run web ใน app ของเราได้ รวมทั้งยังสามารถใช้งานความสามารถต่าง ๆ ของ device ได้ ยกตัวอย่างเช่น กล้องถ่ายรูป และ GPS เป็นต้น

โดยที่ Hybrid app จะสร้างส่วนการติดต่อสื่อสารระหว่าง Web View กับ Native ขึ้น แต่ตอนว่าไม่ได้อยู่ในมาตรฐานของ iOS และ Android แต่เป็น 3-party tool/framework ที่มาครอบอีกชั้นหนึ่งยกตัวอย่างเช่น Codova, Ionic, NativeScript, Xamarin, PhoneGap และ React Native เป็นต้น

แนวทางการเลือกพัฒนาแบบ Hybrid หรือ Native App



การจะตัดสินใจเลือกรูปแบบการพัฒนาแบบใด ต้องลองพิจารณาถึงข้อดีข้อเสียของแนวทางการพัฒนาในแบบต่าง ๆ ดังนี้

	Native	Hybrid
Development Language	Native only	Native and web / web only
Device Access	Complete	Complete
Device Specific Features	High	Moderate
Speed	Very fast	Medium
App Store	Available	Available
Approval Process	Mandatory	Low overhead
Code Portability	None	High
Advanced Graphics	High	Moderate
UI / UX	High	Moderate
Access to Native APIs	High	Moderate
Development Cost	Expensive	Reasonable

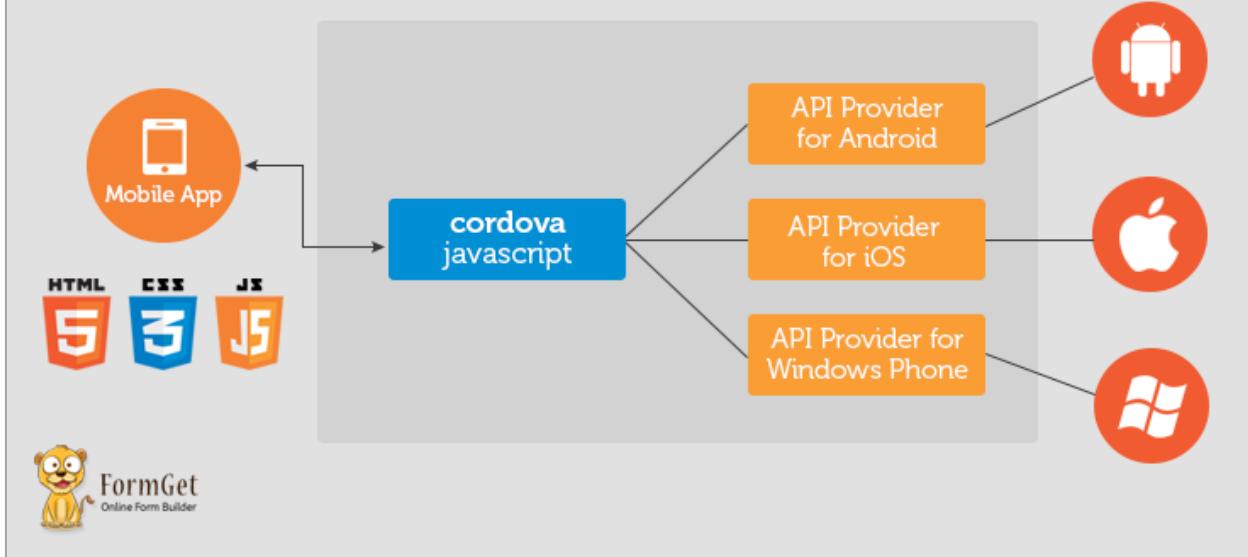
รู้จักกับเทคโนโลยี PhoneGap / Cordova / Ionic



Cordova คือ platform (แพลทฟอร์ม) ตัวหนึ่งที่รับหน้าที่ สร้าง application (แอพพลิเคชั่น) โดยใช้ภาษาHTML (เอช ที อี แอล),CSS3(ซี เอส เอส) และ JS (เจ อส) นั้นก็คือ แค่เราเขียน Web (เว็บ) เป็นเราก็สามารถทำ Application ได้แล้ว ซึ่งถ้าพูดถึงการเขียน App Android (แอพ แอนดรอยด์) จากภาษา Java (จาวา) มันก็ยาก ที่เดียว การใช้ Cordova นั้นเพียงแค่มี Technology (เทคโนโลยี) นี้ช่วยได้ คราวที่เขียนเว็บเก่งอยู่แล้วก็จะทำได้ ง่ายขึ้นความพิเศษของการสร้าง App แบบ Cordova คือเราเขียน Code (โค้ด) แบบเดียวสามารถ Build (บิลด์) เป็น Application ได้แทบทุก OS (ไอ ออส) พุดง่ายๆเลย ก็คือเราเขียน web ครั้งเดียวก็สามารถเลือกได้เลยว่า อย่างให้ออกมาเป็น App ของ OS อะไรก็ได้ ตอนนี้ Cordova สามารถสร้าง App ให้ระบบปฏิบัติการ ไหนได้บ้าง

1. Amazon Fire OS (อเมซอน ไฟ โอ เอส)
 2. Android (แอนดรอยด์)
 3. Bada (บอดี้)
 4. Blackberry (แบล็คเบอร์รี่)
 5. Firefox OS (ไฟฟอก)
 6. iOS (ไอ โอ เอส)
 7. Mac OS X (แม็ค โอ เอส เอ็ก)
 8. Qt (ควี ที)
 9. Tizen (ไทนเซน)
 - 10 Ubuntu (อูบุนตู)
 - 11 Web OS (เว็บ โอ เอส)
 - 12 Windows Phone 7 8 (วินโดว์ โฟน 7 8)

Cordova / PhoneGap



Cordova / PhoneGap ยังมีความสามารถในการติดต่อกับ API ของอุปกรณ์ เช่น กล้องถ่ายภาพ GPS Storage battery และอื่น ๆ ผ่านไฟล์ Cordova.js ทำให้เราสามารถพัฒนาแอปที่เรียกความสามารถในการใช้งาน Sensor ต่าง ๆ ของอุปกรณ์ได้ง่ายขึ้น

รู้จัก ionic framework



ionic framework เป็นเครื่องมือที่ใช้ในการพัฒนา Mobile Application แบบ Hybrid หรือหลายท่าน เคยได้ยินคำว่า "Hybrid App" หรือ "Hybrid Mobile App" คือความสามารถพัฒนาแอปครั้งเดียวแล้วรันได้หลาย

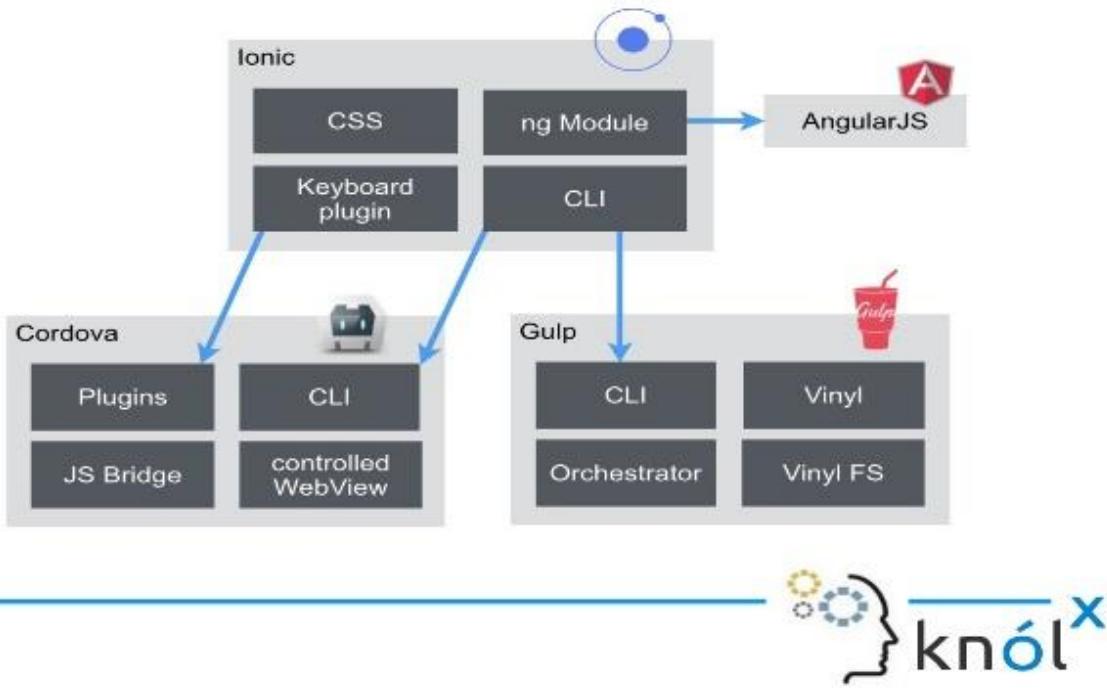
Platform เช่น Android , iOS ที่ได้รับความนิยมในปัจจุบัน ตัว ionic framework เองจะใช้เทคโนโลยีในการพัฒนาคือ HTML5, CSS3 และ JAVA Script ตัวการพัฒนาหลักๆเลยที่เราจะต้องใช้คือ JAVA Script เพราะว่า ionic framework ใช้ Core หลักเป็น Angular JS

ข้อดีที่เด่นๆของ ionic framework ที่เห็นได้ชัดคือ มีความใกล้เคียงกับ Native App มาก ทั้งหน้าตา UI การใช้งาน แต่ก็ไม่ถูกจำกัดเหมือน Native App เลยนะที่เดียว ข้อดีอีกอย่างคือ ionic framework สามารถพัฒนาให้ติดต่อกับ Hardware ของอุปกรณ์ได้ด้วย เช่น กล้อง, ไมโครโฟน การตรวจสอบการเชื่อมต่ออินเทอร์เน็ต เป็นต้น โดยใช้งานร่วมกับ PhoneGap/Cordova

นอกจากนั้นแล้วเรายังสามารถทดสอบแอปที่เราพัฒนาขึ้นผ่านอุปกรณ์จริงได้เลย โดยการทดสอบจาก Application ที่ชื่อว่า ionicview ที่สามารถรองรับทั้ง Android และ iOS โดยใช้คำสั่ง command line ไม่กี่บรรทัด ก็สามารถอัพโหลดแล้วรันได้เลย

Ionic Framework

Ionic is the open source HTML5 Mobile Framework for building amazing, cross-platform hybrid native apps and mobile websites with HTML, JavaScript, and CSS. If you know how to build or design websites, you will be able to build a real mobile app with Ionic!



Module 2: การติดตั้งและเตรียมระบบ

เครื่องคอมพิวเตอร์และอุปกรณ์สำหรับอบรม

- Windows Laptop หรือ Apple MacBook
- CPU i5 ขึ้นไป ความเร็วไม่น้อยกว่า 2ghz (รองรับทั้ง Intel และ AMD)
- RAM ไม่น้อยกว่า 4GB (แนะนำ 8GB)
- HDD มีพื้นที่ว่างอย่างน้อย 20GB ขึ้นไป
- ติดตั้ง Windows 7 , 8 ,10 ขึ้นไป (ไม่รองรับ Windows XP) ได้ทั้ง 32bits และ 64bits
- สำหรับ MacBook แนะนำเป็น macos sierra (10.12 ขึ้นไป)
- Smart Phone Android (แนะนำรุ่นที่รองรับตั้งแต่ Android 4.4.x ขึ้นไป) พร้อมสาย data เชื่อมต่อข้อมูล
- Apple Device (Iphone , iPad) หากต้องการทดสอบของ iOS (ไม่มีบังคับ) ที่รองรับ iOS 10 ขึ้นไป
พร้อมสาย data สำหรับเชื่อมต่อข้อมูลกับคอมพิวเตอร์

ระบบและโปรแกรมที่ใช้บน Windows (รองรับเฉพาะการพัฒนา Android)

เครื่องมือเขียนโค้ด (Editor)

- Visual Studio Code and Extensions

Web Server

- XAMPP and Config

Web Browser

- Google Chrome and Extension

เครื่องมือสำหรับการพัฒนา Ionic framework

- Java Development Kit
- Node JS
- Git
- Android Studio และชุด SDK
- Cordova Framework
- Ionic Framework
- USB Driver device

ระบบและโปรแกรมที่จะใช้บน Mac (รองรับการพัฒนาทั้ง Android และ iOS)

เครื่องมือเขียนโค้ด (Editor)

- Visual Studio Code and Extensions

Web Server

- XAMPP and Config

Web Browser

- Google Chrome and Extension

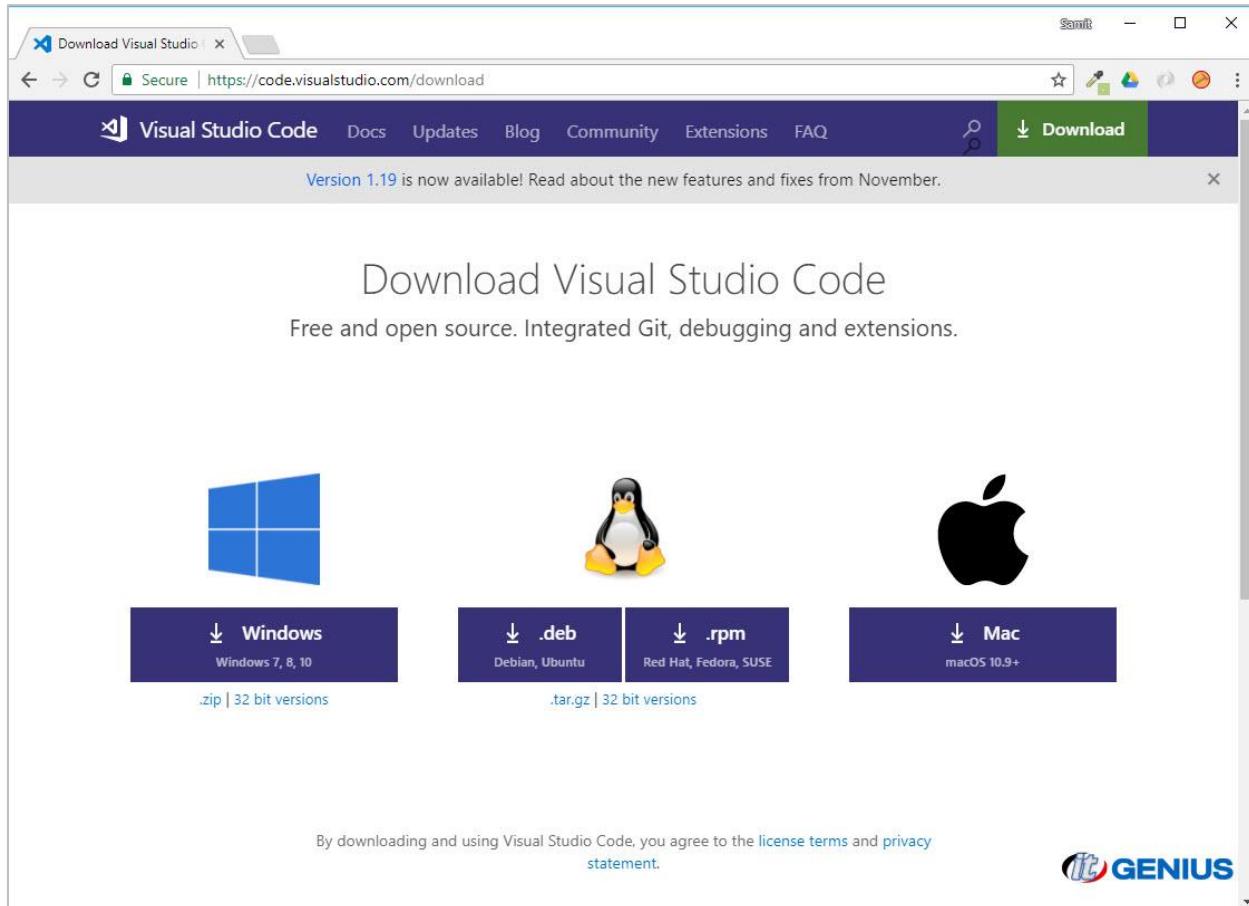
เครื่องมือสำหรับการพัฒนา Ionic framework (Platform Android)

- Java Development Kit
- Node JS
- Git
- Android Studio และชุด SDK
- Cordova Framework
- Ionic Framework
- Homebrew
- gradle

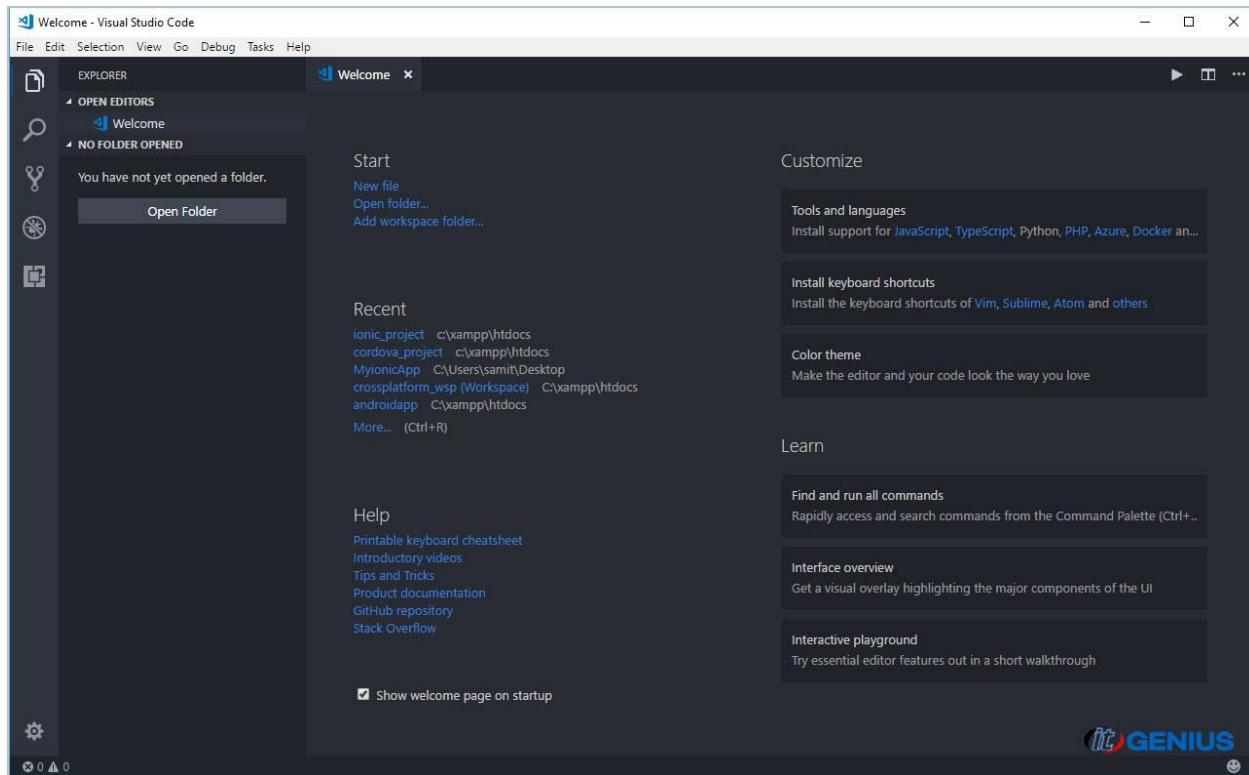
เพิ่มเติมเครื่องมือสำหรับ Platform iOS

- Xcode and Simulator
- ios-sim
- ios-deploy

ขั้นตอนการติดตั้ง Visual Studio Code



เข้าไปดาวน์โหลดไฟล์ติดตั้งได้ที่ <https://code.visualstudio.com/> เลือกให้ตรงกับระบบปฏิบัติการของท่าน
(รองรับทั้ง windows , linux และ mac osx)
เมื่อติดตั้งเสร็จแล้วทำการเปิดขึ้นมาดังภาพ

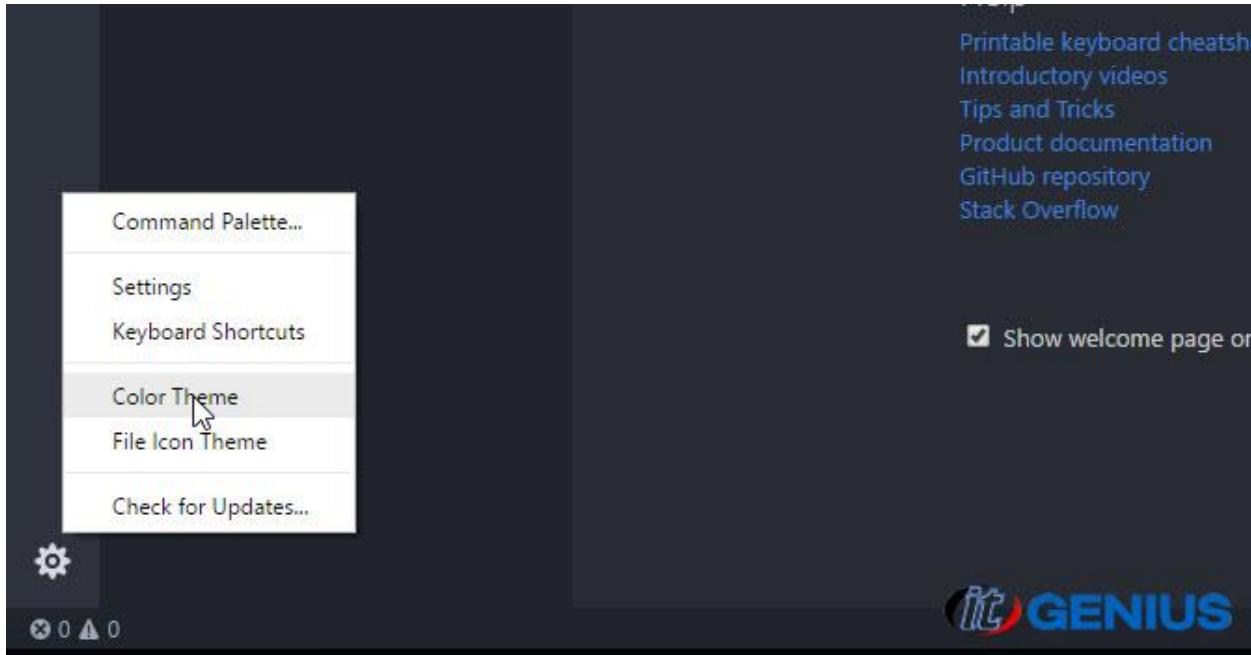


ขั้นตอนการติดตั้งส่วนเสริม (Extensions)

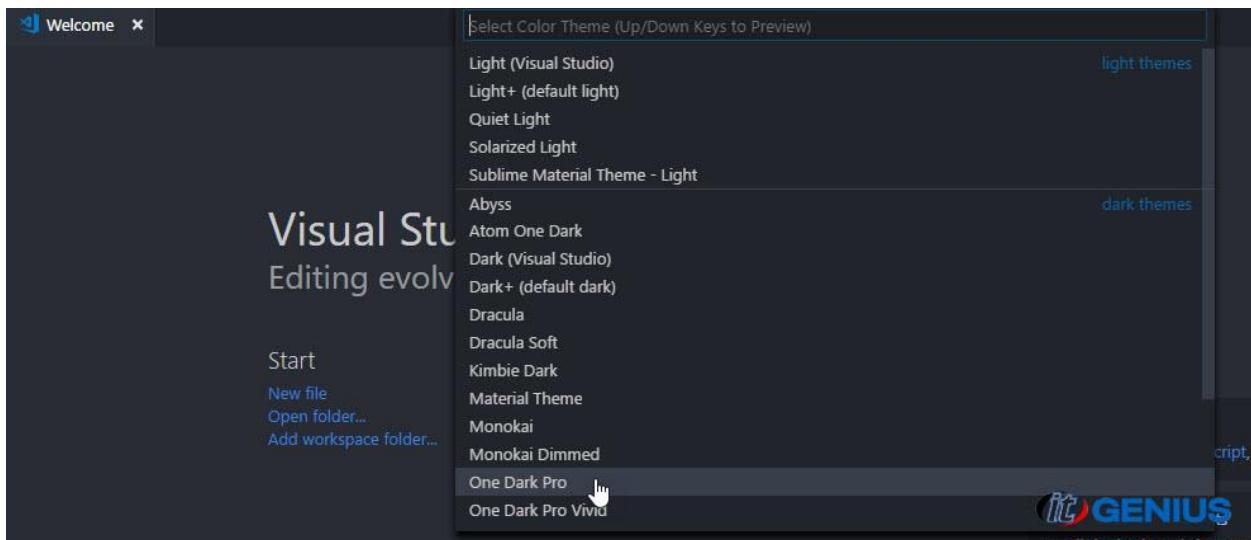
การติดตั้งส่วนเสริมจะทำให้ Visual Studio Code ใช้งานได้สะดวก และง่ายต่อการพัฒนาโปรแกรมต่าง ๆ เพราะจะมีตัวช่วยในการแนะนำคำสั่งต่าง ๆ พร้อมช่วยจัดรูปแบบโค้ดคำสั่งให้เรียบร้อยสวยงามขึ้น โดยรายการส่วนเสริมที่แนะนำมีดังนี้

- | | |
|---|---|
| <ol style="list-style-type: none"> 1. Beautify by HookyQR 2. VS Color Picker by lihui 3. AutoFileName by JerryHong 4. highlight-matching-tag by vincaslt 5. vscode-icons by Roberto Huertas 6. Sass by Robin Bentley 7. JavaScript (ES6) code snip by charalampos karypidi | <ol style="list-style-type: none"> 8. Angular v6 Snippets by John Papa 9. Ionic 2 snippets by oudZy 10. Ionic 3 ionView Snippets by Daniel Ehrhardt 11. Ionic 3 snippets by Satit Rianpi 12. One Dark Pro by zhuangtongf |
|---|---|

การปรับแต่ง Theme



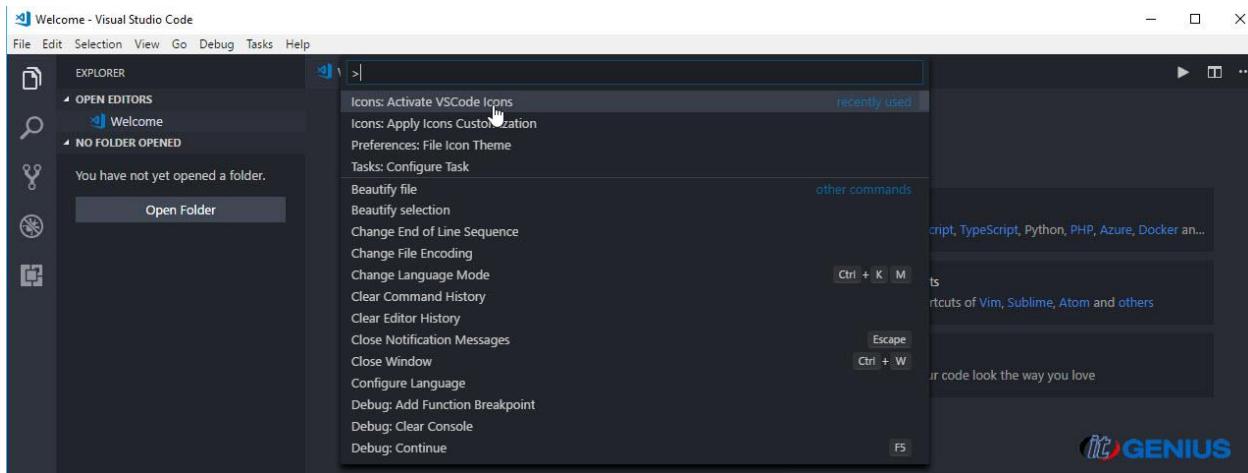
คลิกที่เมนู File > Preference > Color Theme หรือเลือกจากเมนูดังภาพก็ได้



จากนั้นคลิกเลือก Theme ที่ต้องการได้เลย

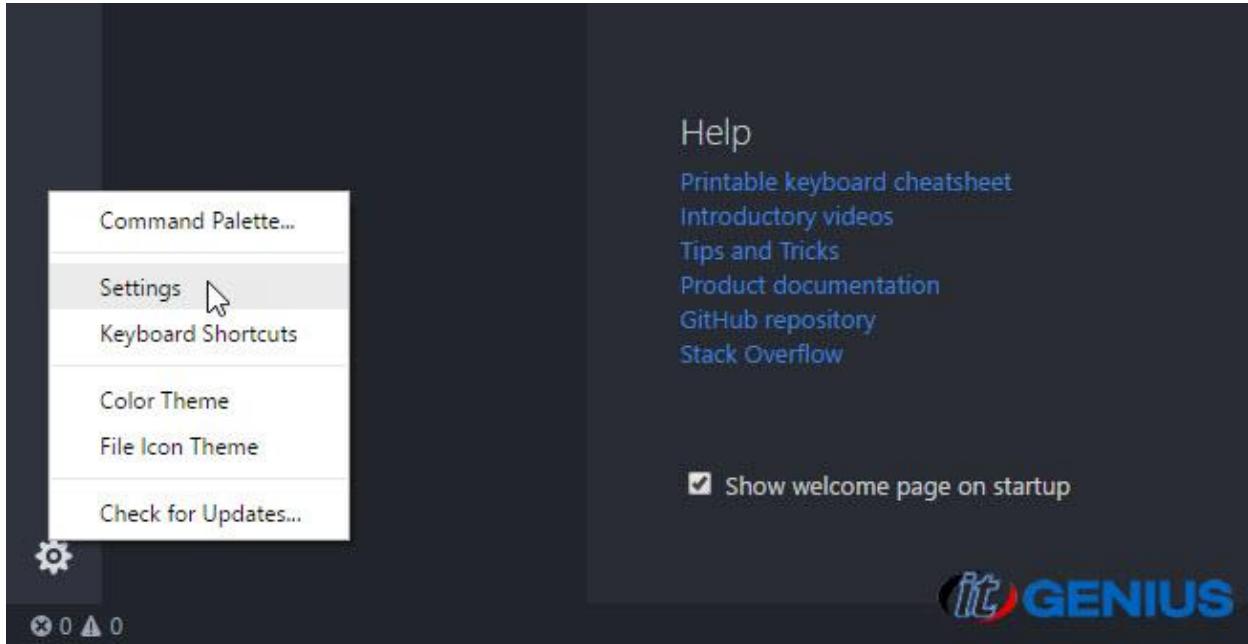
การแสดง Icon File ในโปรแกรม VS Code

กดแป้น Ctrl+Shift+P และจะปรากฏหน้าต่างดังภาพ

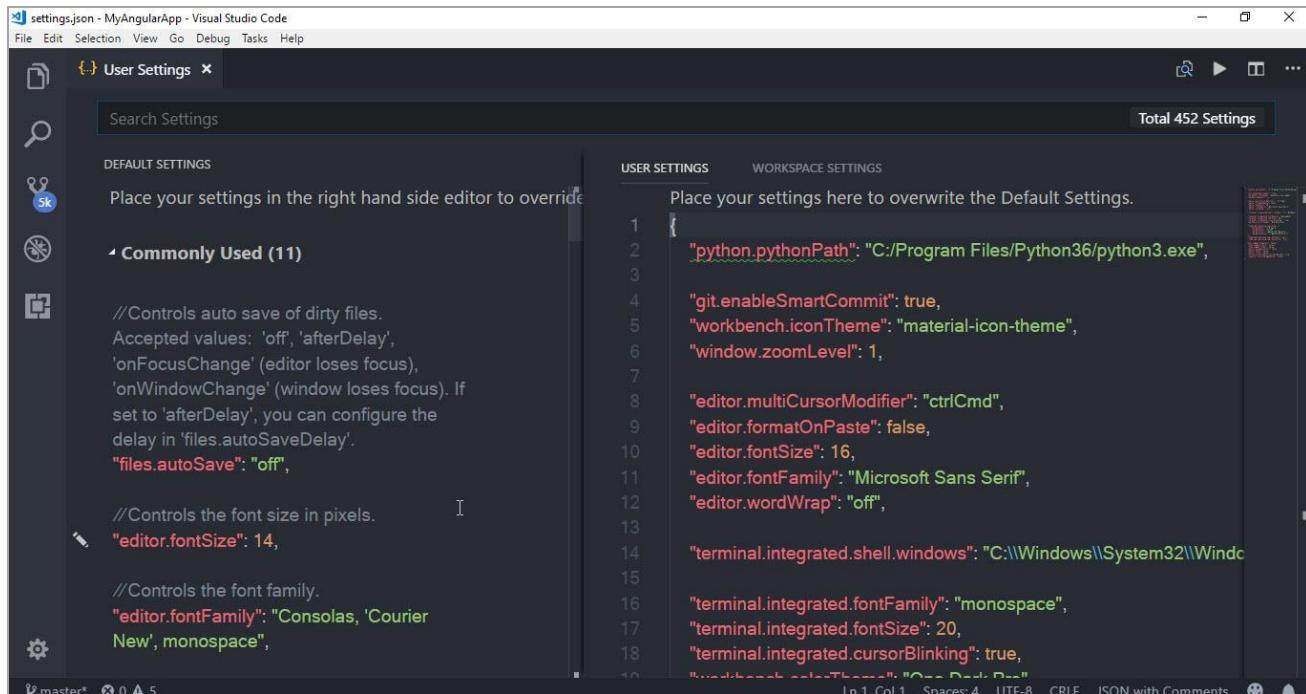


พิมพ์ค้นหาว่า “icon” และคลิกเลือก Icons: Activate VSCode Icons ดังภาพ

การปรับแต่งค่าเริ่มต้นของโปรแกรม



คลิกที่เมนู File > Preference > Settings หรือเลือกจากเมนูดังภาพก็ได้



```

settings.json - MyAngularApp - Visual Studio Code
File Edit Selection View Go Debug Tasks Help
User Settings x
Search Settings
Total 452 Settings
DEFAULT SETTINGS
Place your settings in the right hand side editor to override the default settings.
Commonly Used (11)
//Controls auto save of dirty files.
Accepted values: 'off', 'afterDelay',
'onFocusChange' (editor loses focus),
'onWindowChange' (window loses focus). If
set to 'afterDelay', you can configure the
delay in 'files.autoSaveDelay'.
"files.autoSave": "off",

//Controls the font size in pixels.
"editor.fontSize": 14,

//Controls the font family.
"editor.fontFamily": "Consolas, 'Courier
New', monospace",
USER SETTINGS
WORKSPACE SETTINGS
Place your settings here to overwrite the Default Settings.

{
  "python.pythonPath": "C:/Program Files/Python36/python3.exe",
  "git.enableSmartCommit": true,
  "workbench.iconTheme": "material-icon-theme",
  "window.zoomLevel": 1,
  "editor.multiCursorModifier": "ctrlCmd",
  "editor.formatOnPaste": false,
  "editor.fontSize": 16,
  "editor.fontFamily": "Microsoft Sans Serif",
  "editor.wordWrap": "off",
  "terminal.integrated.shell.windows": "C:\\Windows\\System32\\Wind
  "terminal.integrated.fontFamily": "monospace",
  "terminal.integrated.fontSize": 20,
  "terminal.integrated.cursorBlinking": true,
  "terminal.integrated.tabsFontSize": 14
}
Ln 1, Col 1  Spaces: 4  UTF-8  CRLF  JSON with Comments

```

เข้าไปทำการคัดลอกคำสั่งการตั้งค่าได้ที่ bit.ly/vscodesetting (เฉพาะบน Windows) จากนั้นนำคำสั่งทั้งหมดมาวางที่ไฟล์ settings.json ดังภาพ ทำการบันทึกไฟล์ก็จะสามารถใช้งานได้

ติดตั้ง XAMPP

ในการอบรมครั้งนี้ จะมีการทดสอบเขียนและพิเคราะห์ API จากภายนอก ดังนั้นเพื่อให้สะดวกต่อ การทดสอบ จึงแนะนำให้จำลองเครื่องเป็น Web Server ที่สามารถรันสคริปต์ผ่าน Server ได้ เช่นภาษา PHP , ASP, JSP, Node JS เป็นต้น ซึ่งในที่นี้เลือกใช้ภาษา PHP จึงแนะนำขั้นตอนการติดตั้ง XAMPP เพื่อจำลองเป็น Web Server ดังนี้

The screenshot shows a web browser window with the URL <https://www.apachefriends.org/download.html>. The page title is "Download". A sub-section titled "XAMPP for Windows" displays two download options:

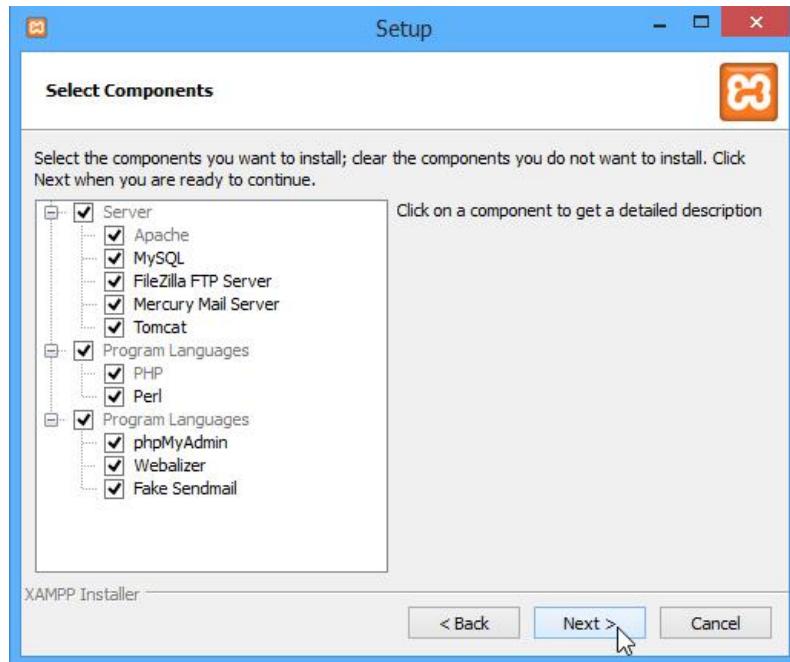
Version	Checksum	Size
5.6.33 / PHP 5.6.33	What's Included? md5 sha1	Download (32 bit) 109 Mb
7.0.27 / PHP 7.0.27	What's Included? md5 sha1	Download (32 bit) 120 Mb

To the right, a "Documentation/FAQs" sidebar provides links to various FAQ sections.

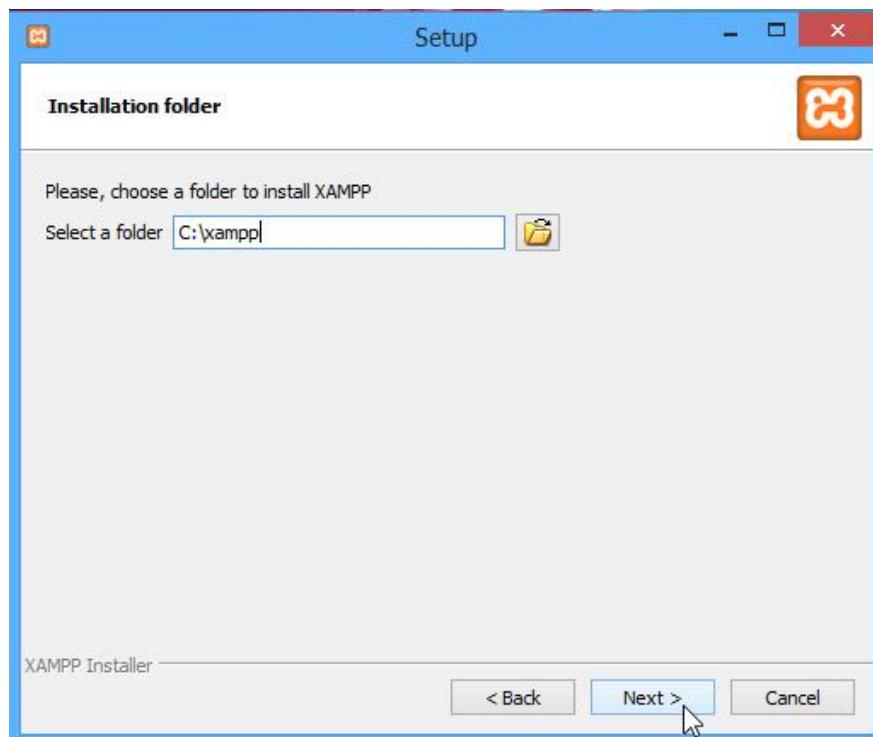
เข้าไปดาวน์โหลด XAMPP ได้ที่ลิงก์ <https://www.apachefriends.org/download.html> เลือก Version ของ PHP เป็น 5.6.xxx ขึ้นไป ดังภาพ



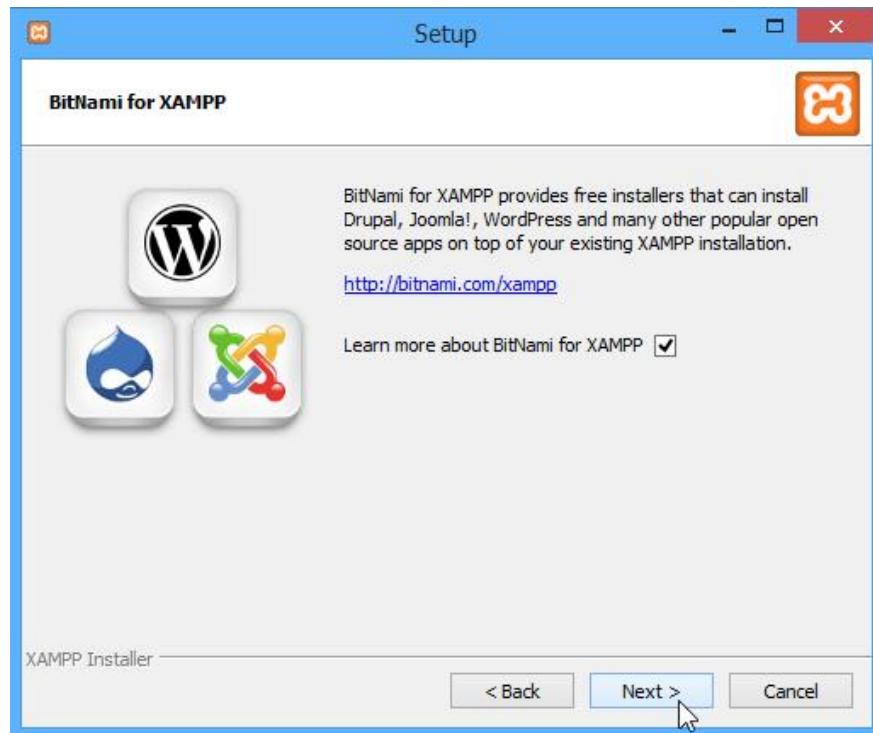
หลังจากดาวน์โหลดมาแล้ว เลือกคลิกติดตั้งตามรูป คลิกปุ่ม Next



เลือกส่วนประกอบต่าง ๆ ดังรูปแล้วคลิก Next



ใช้ค่าเริ่มต้นการติดตั้งไว้ที่ C:\xampp ดังภาพ



คลิกปุ่ม Next เพื่อทำการติดตั้ง



รอการติดตั้ง ...



เมื่อติดตั้งเสร็จคลิกปุ่ม Finish

The screenshot shows the XAMPP Control Panel v3.2.2 interface. The title bar reads "XAMPP Control Panel v3.2.2 [Compiled: Nov 12th 2015]". The main area displays a table of services:

Service	Module	PID(s)	Port(s)	Actions			
	Apache	12216 11964	80, 443	Stop (highlighted with a cursor)	Admin	Config	Logs
	MySQL	11976	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

To the right of the table is a vertical toolbar with icons for Config, Netstat, Shell, Explorer, Services, Help, and Quit. Below the table is a log window showing the following messages:

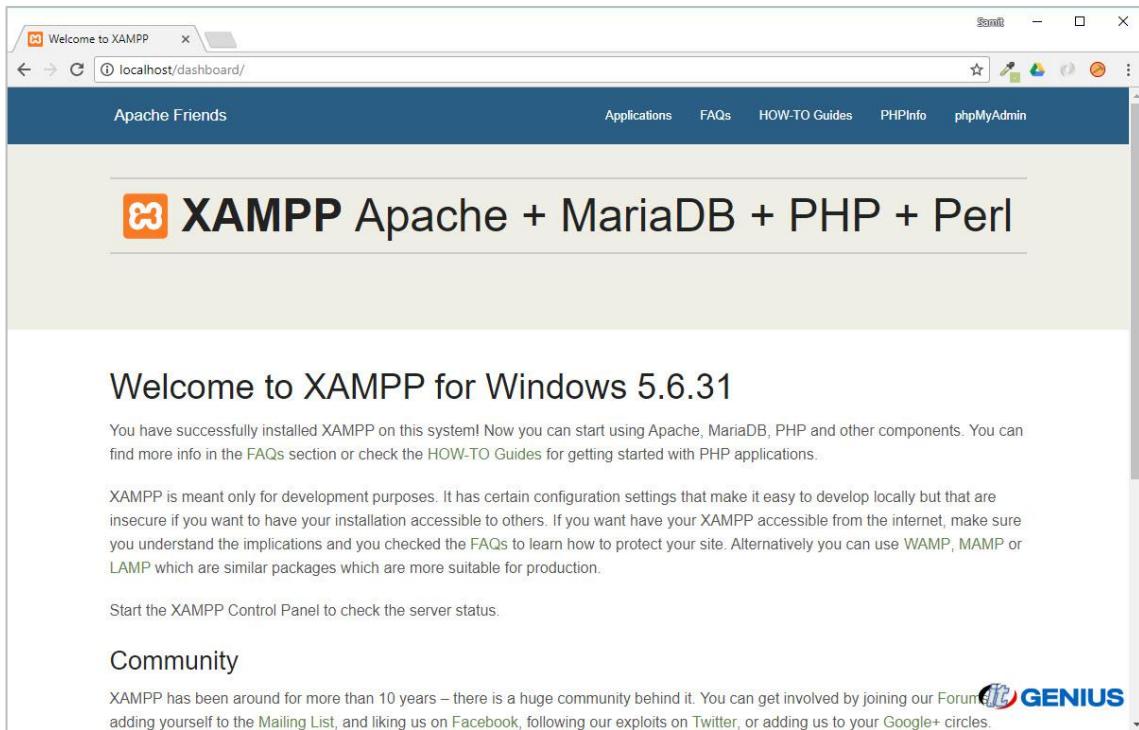
```

11:40:11 PM [main] Control Panel Ready
11:40:12 PM [Apache] Autostart active: starting...
11:40:12 PM [Apache] Attempting to start Apache app...
11:40:12 PM [mysql] Autostart active: starting...
11:40:12 PM [mysql] Attempting to start MySQL app...
11:40:13 PM [Apache] Status change detected: running
11:40:13 PM [mysql] Status change detected: running

```

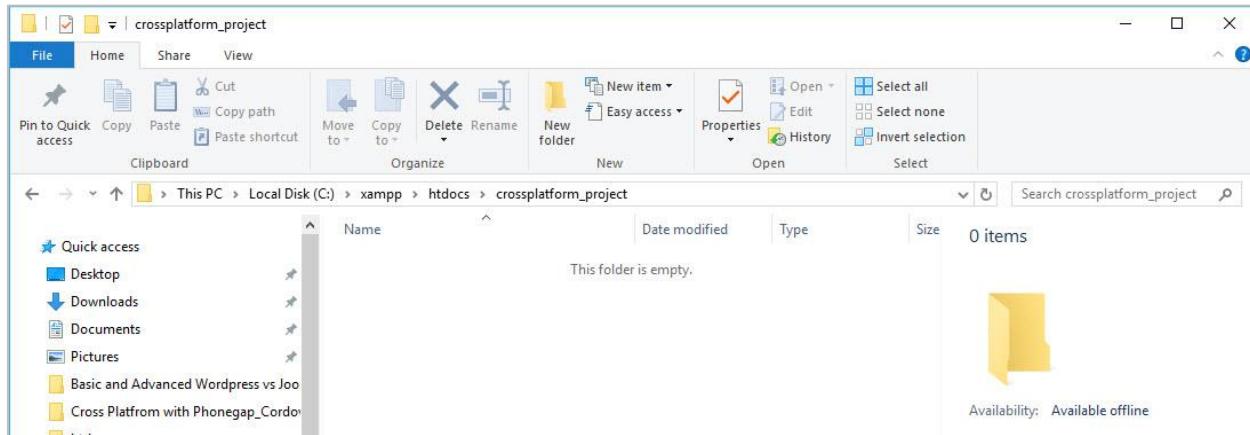
At the bottom right of the control panel window is a "GENIUS" logo.

คลิกปุ่ม Start ของ Apache และ MySQL ขึ้นมาเพื่อเริ่มใช้งาน แล้วกดปิดหน้านี้ไปได้

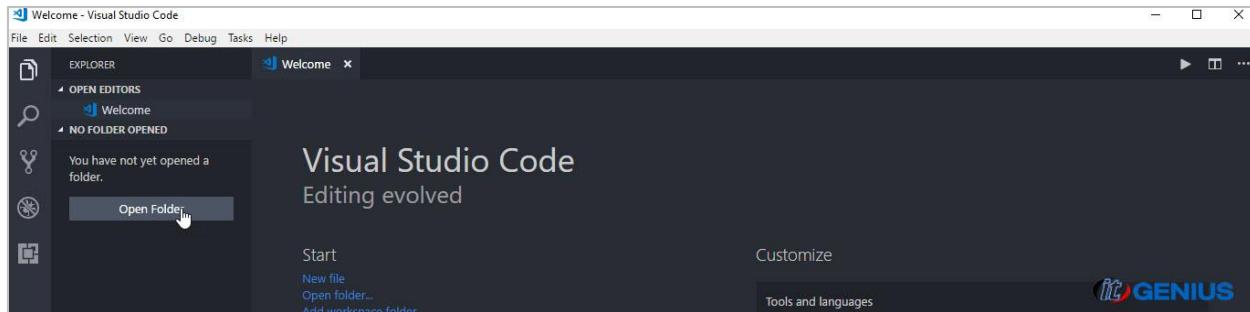


จากนั้นเปิดเว็บбраузரขึ้นมาแล้วพิมพ์ URL ทดสอบ <http://localhost> หากพบหน้าจอดังภาพนี้ แสดงว่าสามารถทำงานได้

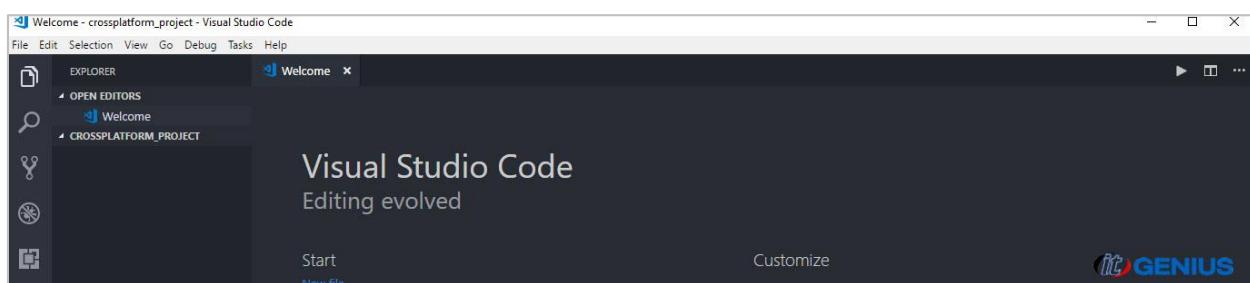
สร้างโปรเจกต์ใหม่ใน XAMPP



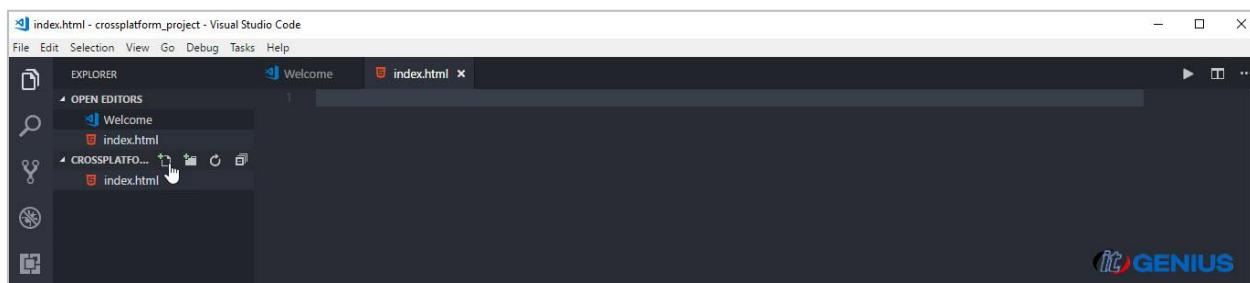
เข้าไปที่ C:\xampp\htdocs จากนั้นทำการสร้างโฟล์เดอร์เพื่อเก็บโปรเจกต์ที่จะฝึกสร้างในคอร์สอบรมครั้งนี้ ชื่อ crossplatform_project ดังภาพ



จากนั้นเปิดโปรแกรม Visual Studio Code ขึ้นมา และทำการเปิดโฟล์เดอร์ crossplatform_project เข้ามา โดยคลิกที่เมนู Open Folder ดังภาพ



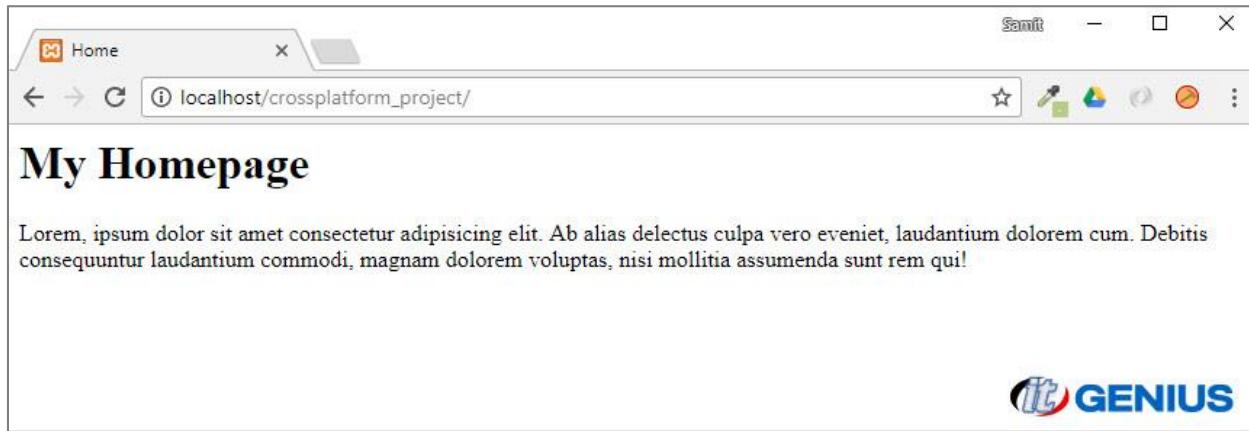
หน้าจอแสดงเมื่อเปิดโฟล์เดอร์เข้ามาแล้ว



ทดสอบสร้างไฟล์ใหม่ คลิกที่ไอคอน New File ดังรูปแล้วกำหนดชื่อ index.html



จากนั้นลองเพิ่มคำสั้ง html ดังภาพ



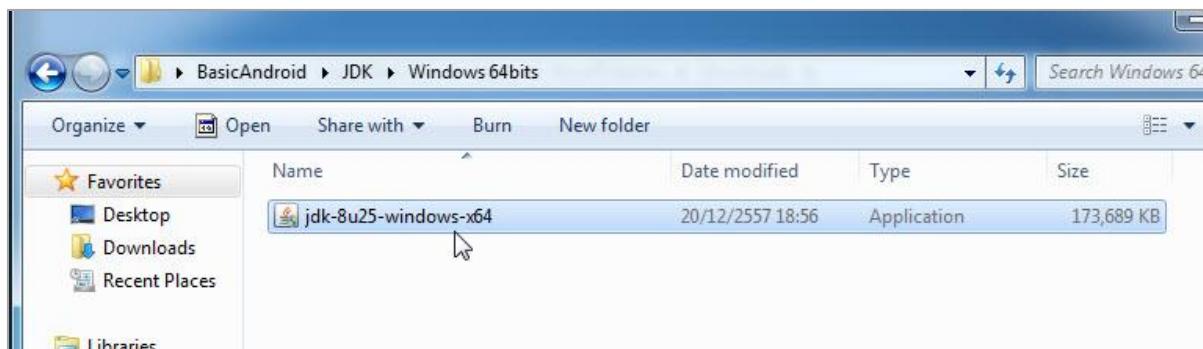
เปิดบราวเซอร์ขึ้นมาพิมป์เข้าไปที่ http://localhost/crossplatform_project น่าจะได้ผลลัพธ์ดังภาพ แสดงว่าตัว Webserver ของ XAMPP ทำงานได้ปกติ และเว็บไซต์เรารสามารถใช้งานได้

วิธีติดตั้ง JDK (Java Development Kit)

Product / File Description	File Size	Download
Linux x64	155.24 MB	jdk-8u25-linux-x64.rpm
Linux -64	154.98 MB	jdk-8u25-linux-x64.tar.gz
Linux x64	135.6 MB	jdk-8u25-linux-x64.rpm
Linux x64	153.42 MB	jdk-8u25-linux-x64.tar.gz
Mac OS X x64	209.13 MB	jdk-8u25-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	137.01 MB	jdk-8u25-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	97.14 MB	jdk-8u25-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	137.11 MB	jdk-8u25-solaris-x64.tar.Z
Solaris x64	94.74 MB	jdk-8u25-solaris-x64.tar.gz
Windows x64	157.26 MB	jdk-8u25-windows-x64.exe
Windows x64	169.62 MB	jdk-8u25-windows-x64.exe

เข้าไปดาวน์โหลด Java SE ได้ที่

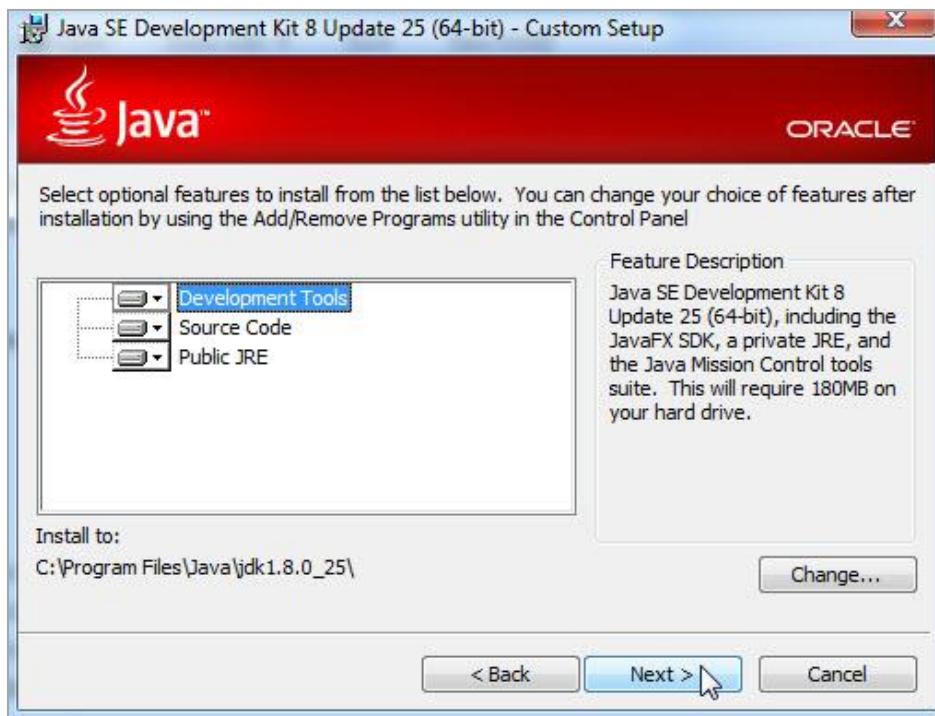
<http://www.oracle.com/technetwork/java/javase/downloads/>



ตัวอย่างไฟล์ Java JDK ที่ดาวน์โหลดมา



คลิกปุ่ม Next เพื่อเริ่มทำการติดตั้ง



เลือก Path สำหรับการติดตั้ง และนำใช้ค่า default ได้เลย



ยืนยันตำแหน่ง path สำหรับการติดตั้ง คลิก Next



รอการติดตั้งสักครู่...



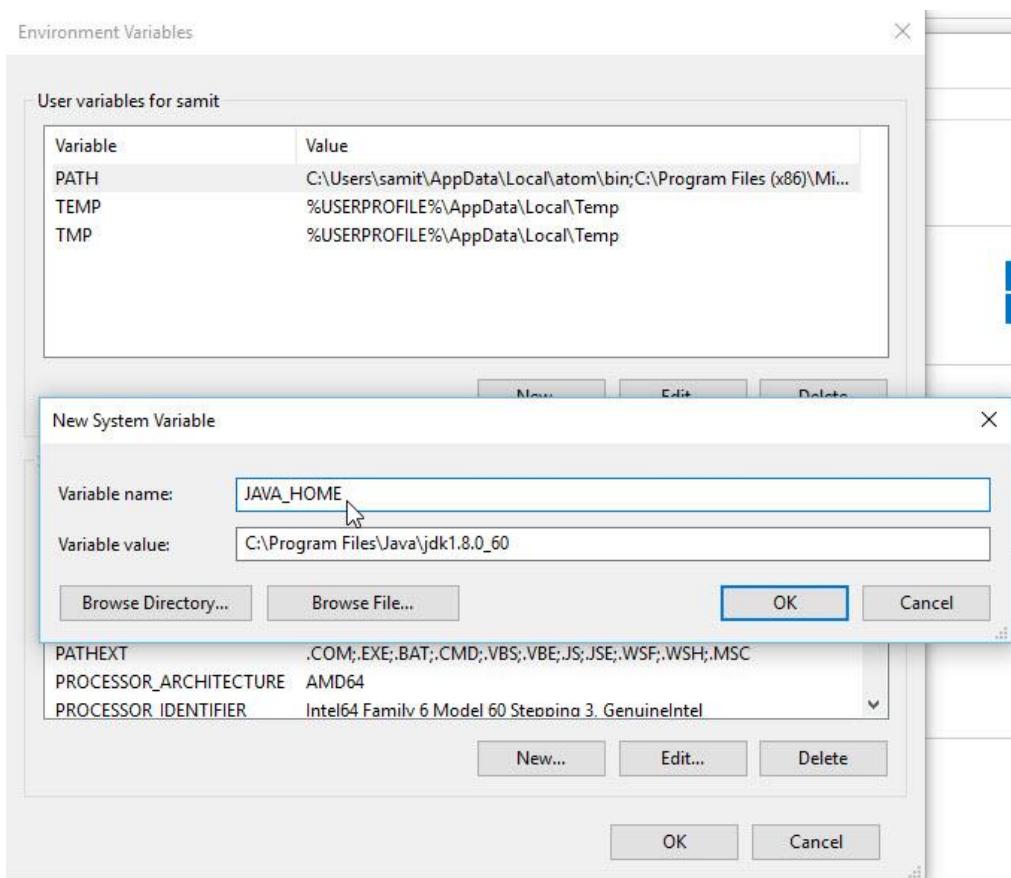
เมื่อการติดตั้งเสร็จสมบูรณ์จะขึ้นหน้าจอหนึ่ง ทำการคลิกปุ่ม Close ปิดหน้าจอหนึ่งไป

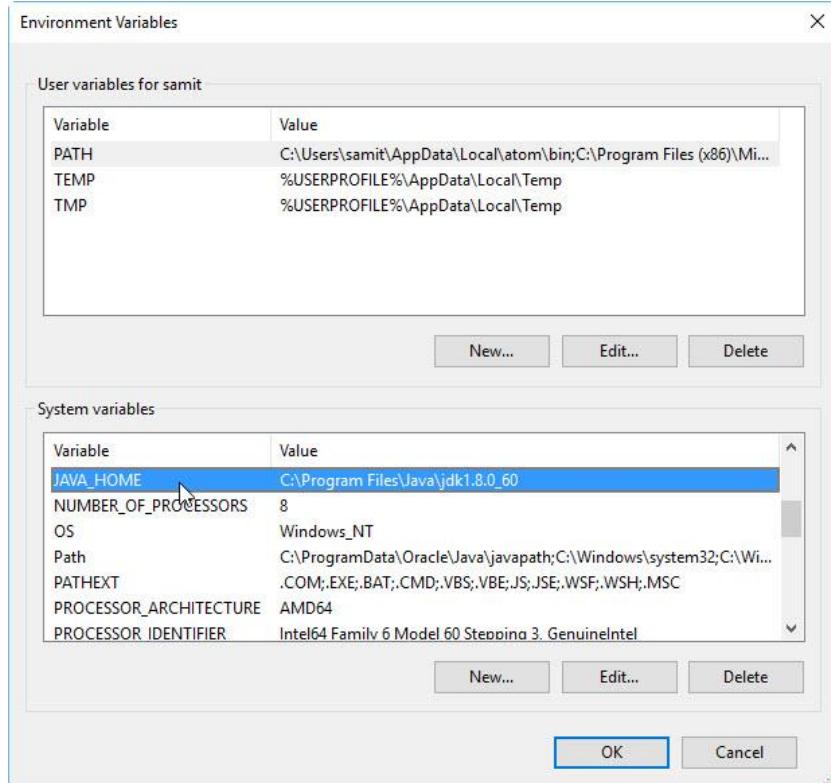
```
C:\>java -version
java version "1.8.0_25"
Java(TM) SE Runtime Environment (build 1.8.0_25-b18)
Java HotSpot(TM) 64-Bit Server VM (build 25.25-b02, mixed mode)

C:\>
```

หรือเข้า Console แล้วเรียก **java -version**
จะพบว่ามี Java SDK ถูกติดตั้งไว้เรียบร้อยแล้ว

การกำหนด (config) Path ของ JDK





เมื่อกำหนด path JAVA_HOME เรียบร้อยแล้วคลิก OK ออกไป

```
C:\Users\samit>set JAVA_HOME  
JAVA_HOME=C:\Program Files\Java\jdk1.8.0_60  
C:\Users\samit>
```

เข้า Command Prompt พิมพ์คำสั่ง set JAVA_HOME หากมีการแสดง Path กลับมาแสดงว่าถูกต้อง

Tip:: การเช็ต JAVA_HOME บนระบบปฏิบัติการ Mac OSX



โดยปกติบนระบบปฏิบัติการ mac osx เราไม่จำเป็นต้องทำการ set JAVA_HOME ก็สามารถใช้งานได้ปกติ แต่หากผู้อุปกรณ์ต้องการตั้งค่าสามารถทำได้ดังนี้

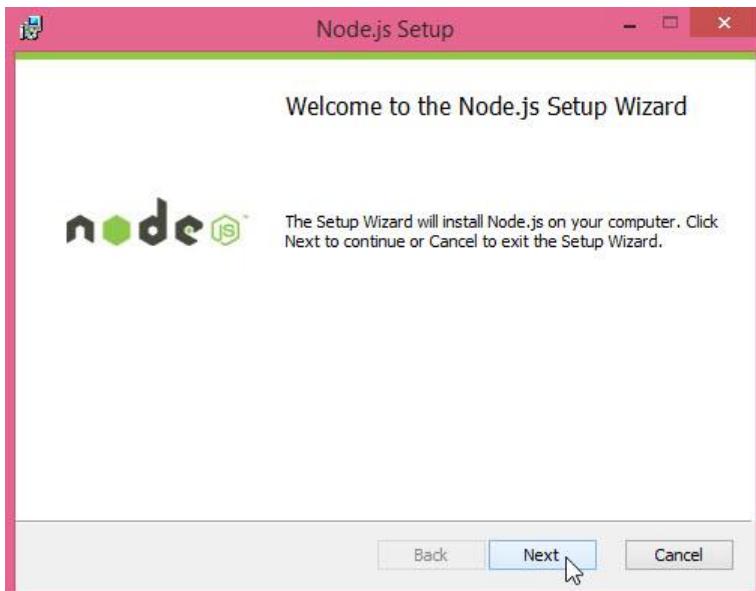
1. เปิดไฟล์ `~/.bash_profile` ในโฟล์เดอร์ user ปัจจุบัน (root user)
2. เพิ่มคำสั่ง `export JAVA_HOME=$(/usr/libexec/java_home)` และบันทึกไฟล์
3. เปิด Terminal ทดสอบป้อนคำสั่งตรวจสอบ `echo $JAVA_HOME` หากถูกต้องจะแสดงผลลัพธ์เป็น path ออกมาดังนี้ `/Library/Java/JavaVirtualMachines/jdk1.8.0_111.jdk/Contents/Home`

วิธีติดตั้ง Node JS บน Windows และ Mac

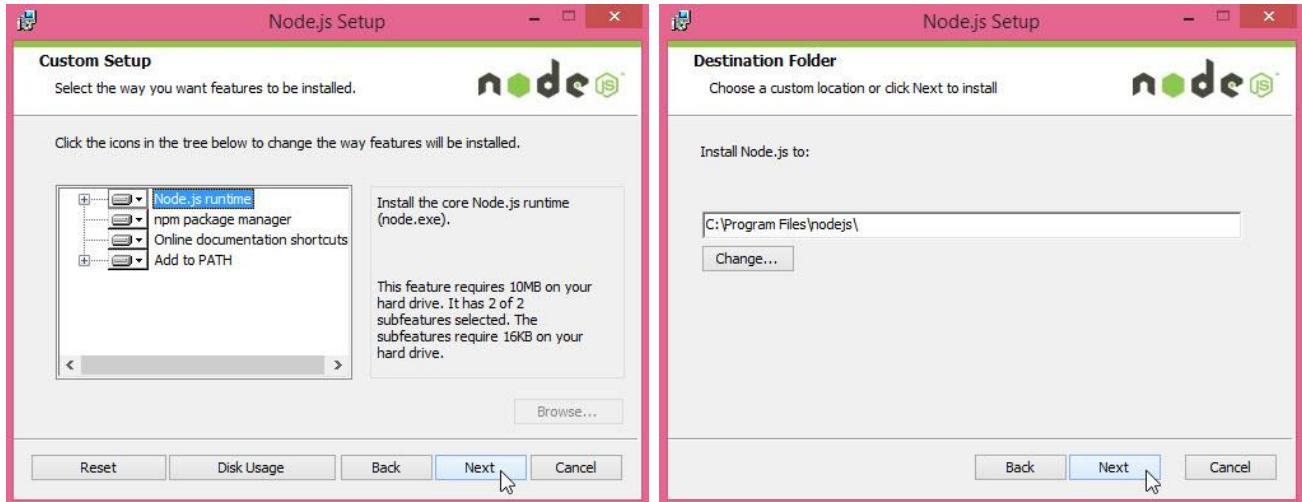
Node JS ถือเป็นเครื่องมือสำคัญในการสร้างแอพพลิเคชันแบบ Cross Platform เพราะถือเป็นตัวกลางในการติดตั้งเครื่องมือ ดาวน์โหลด Library ต่าง ๆ มาลงในโปรเจกต์ของเรา ดังนั้นก่อนจะติดตั้งส่วนอื่น ๆ เราจำเป็นต้องติดตั้ง NodeJS เสียก่อนดังนี้

The screenshot shows the official Node.js website at <https://nodejs.org/en/>. The page features the Node.js logo and navigation links for HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, NEWS, and FOUNDATION. A green banner at the top states "Node 4.x is End Of Life – April Release Updates". Below it, a section for "Download for Windows (x64)" offers two options: "8.11.2 LTS" (Recommended For Most Users) and "10.1.0 Current" (Latest Features). Both options include links to "Other Downloads", "Changelog", and "API Docs". A note below says "Or have a look at the [Long Term Support \(LTS\)](#) schedule." At the bottom, there's a link to "Sign up for Node.js Everywhere, the official Node.js Weekly Newsletter." The URL in the browser address bar is <https://nodejs.org/dist/v8.11.2/node-v8.11.2-x64.msi>.

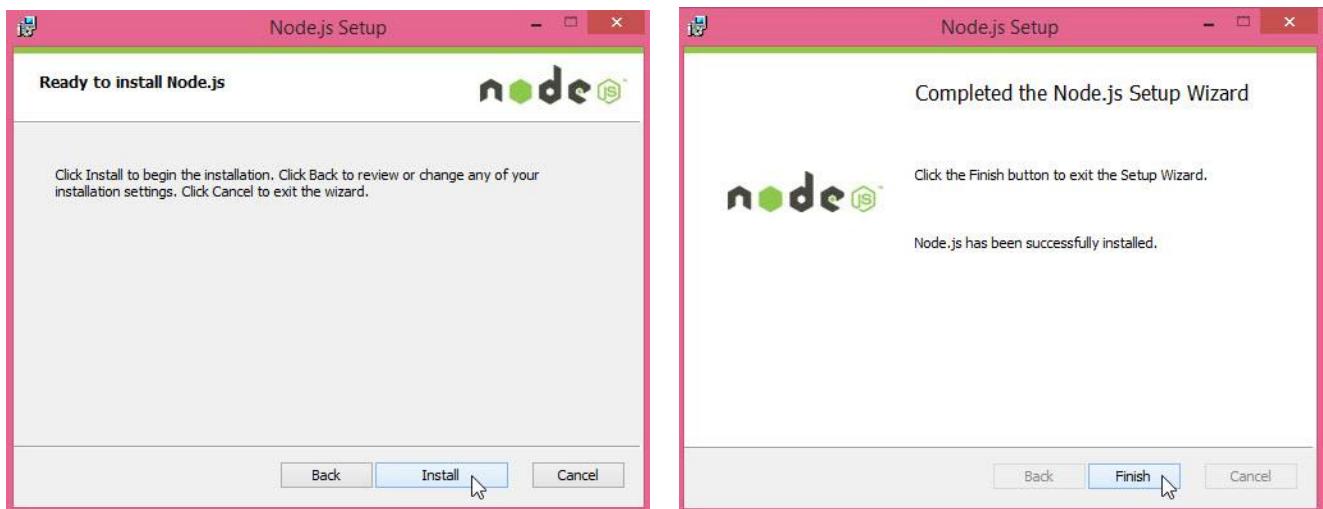
เข้าเว็บไซต์ <https://nodejs.org/en/> เพื่อดาวน์โหลด NodeJS (แนะนำเวอร์ชัน 8.xx) มาลงในเครื่องผู้ใช้



เริ่มติดตั้งจากไฟล์ที่ดาวน์โหลดมา คลิก Next ไปยังขั้นตอนต่อไป



เลือกติดตั้งไว้ที่ path ที่ต้องการ (แนะนำเป็นค่าเริ่มต้นได้เลย)



คลิกปุ่ม Install และรอจนแล้วเสร็จ ... แล้วคลิกปุ่ม Finished เป็นอันเสร็จการติดตั้ง NodeJS แบบง่าย

```
C:\WINDOWS\system32\cmd.exe
C:\Users\ samit>node -v
v8.11.1

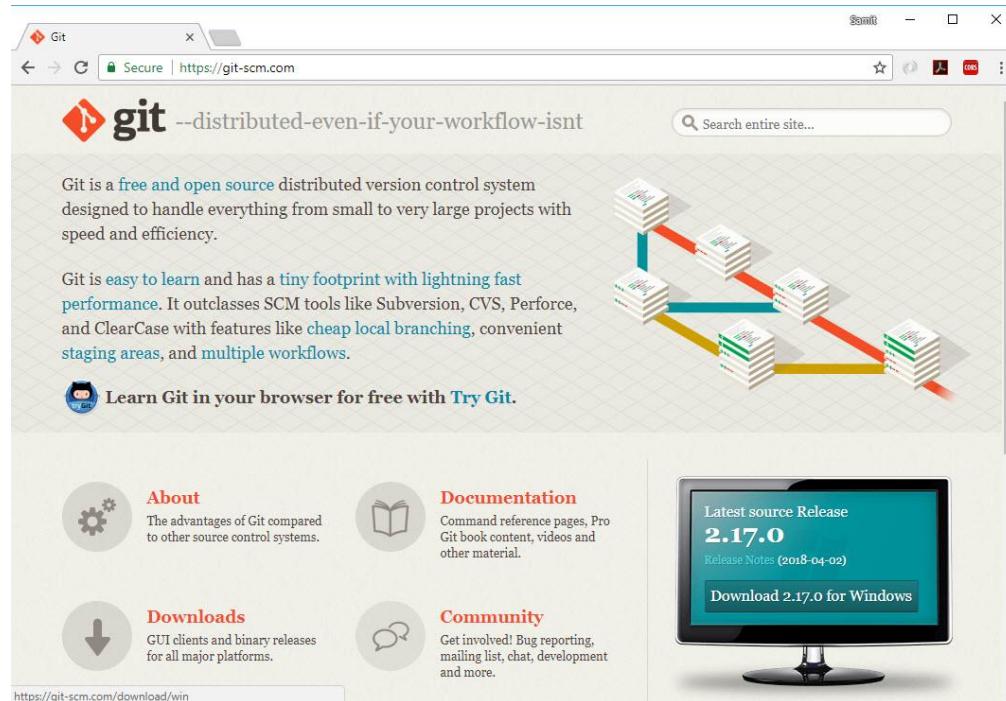
C:\Users\ samit>npm -v
5.6.0

C:\Users\ samit>
```

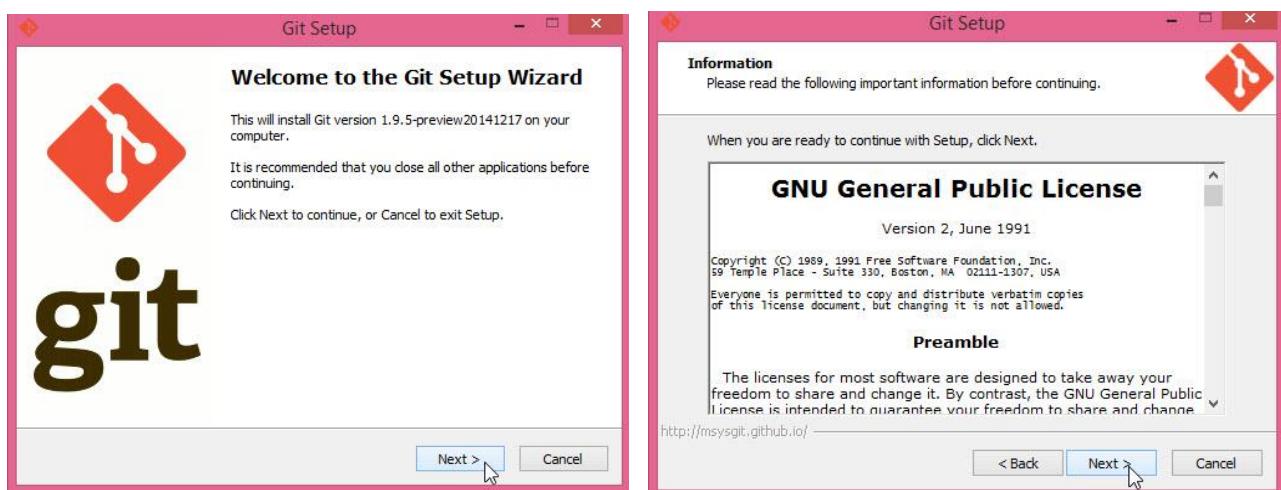
เปิด Command Prompt พิมพ์คำสั่ง node -v และ npm -v หากแสดงเวอร์ชันมาก็แสดงว่าถูกต้อง

วิธีติดตั้ง Git และกำหนดค่าเริ่มต้น สำหรับ Windows และ Mac

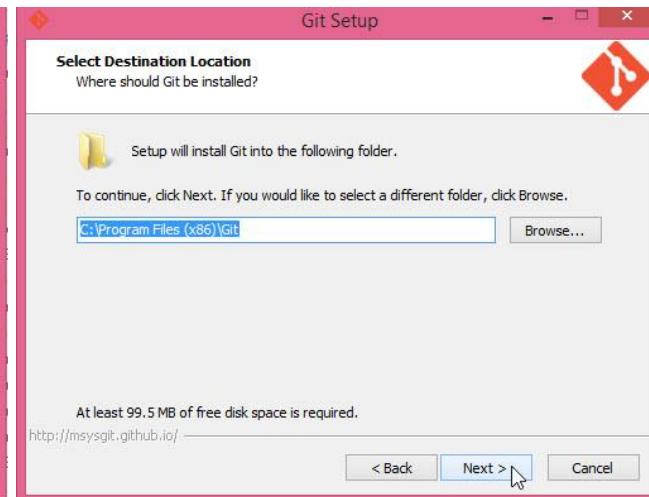
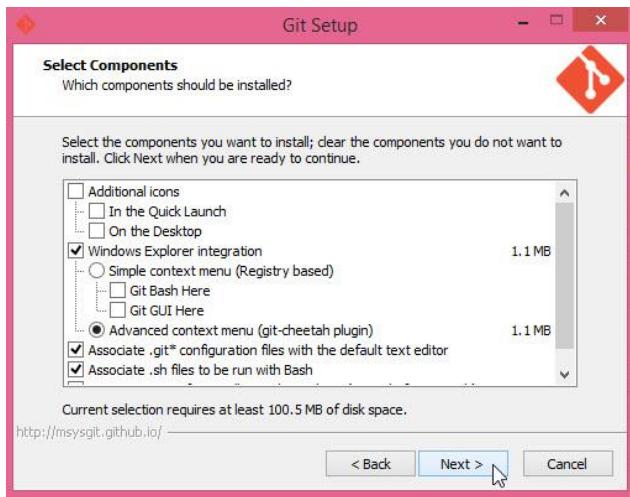
Git เป็นเครื่องมือสำหรับการเก็บประวัติการทำงานของโค้ดเรา ช่วยให้เราสามารถ แล่นนำโค้ดไปแบ่งปัน รวมทั้งทำงานกันเป็นทีมได้สะดวกยิ่งขึ้น โดยในการอบรมครั้งนี้เราจะใช้ git ในการอัพไฟล์และส่งต่อไฟล์กัน



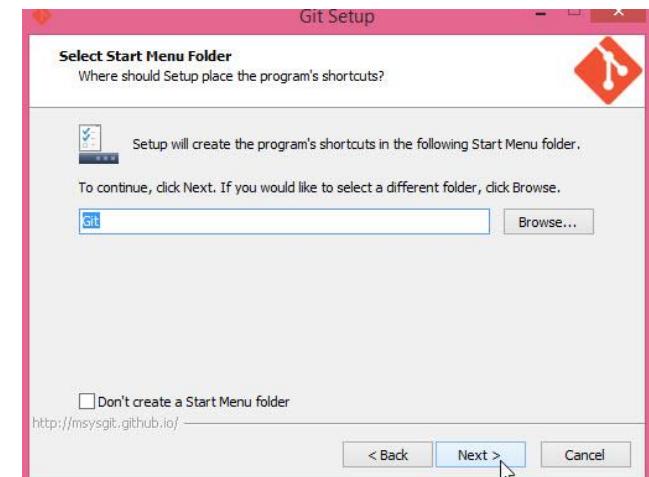
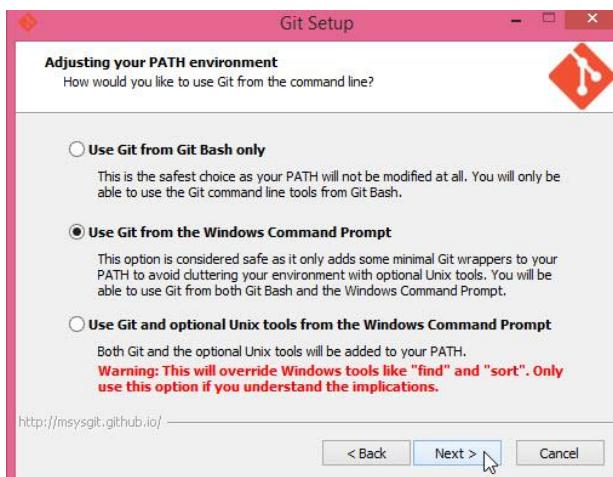
เข้าเว็บไซต์ <https://git-scm.com/> เพื่อดownload ดาวน์โหลดโปรแกรม git (มีให้เลือกทั้งของ Windows และ Mac) จากนั้นทำการดาวน์โหลดลงในเครื่องผู้อบรมไว้



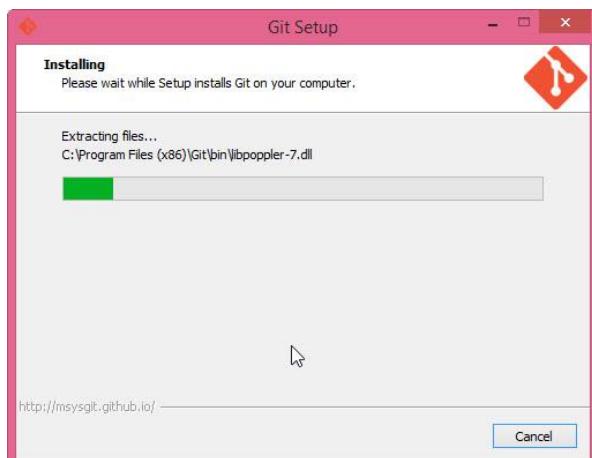
คลิก next เริ่มทำการติดตั้ง



เลือกติดตั้งตามค่าเริ่มต้นได้เลย



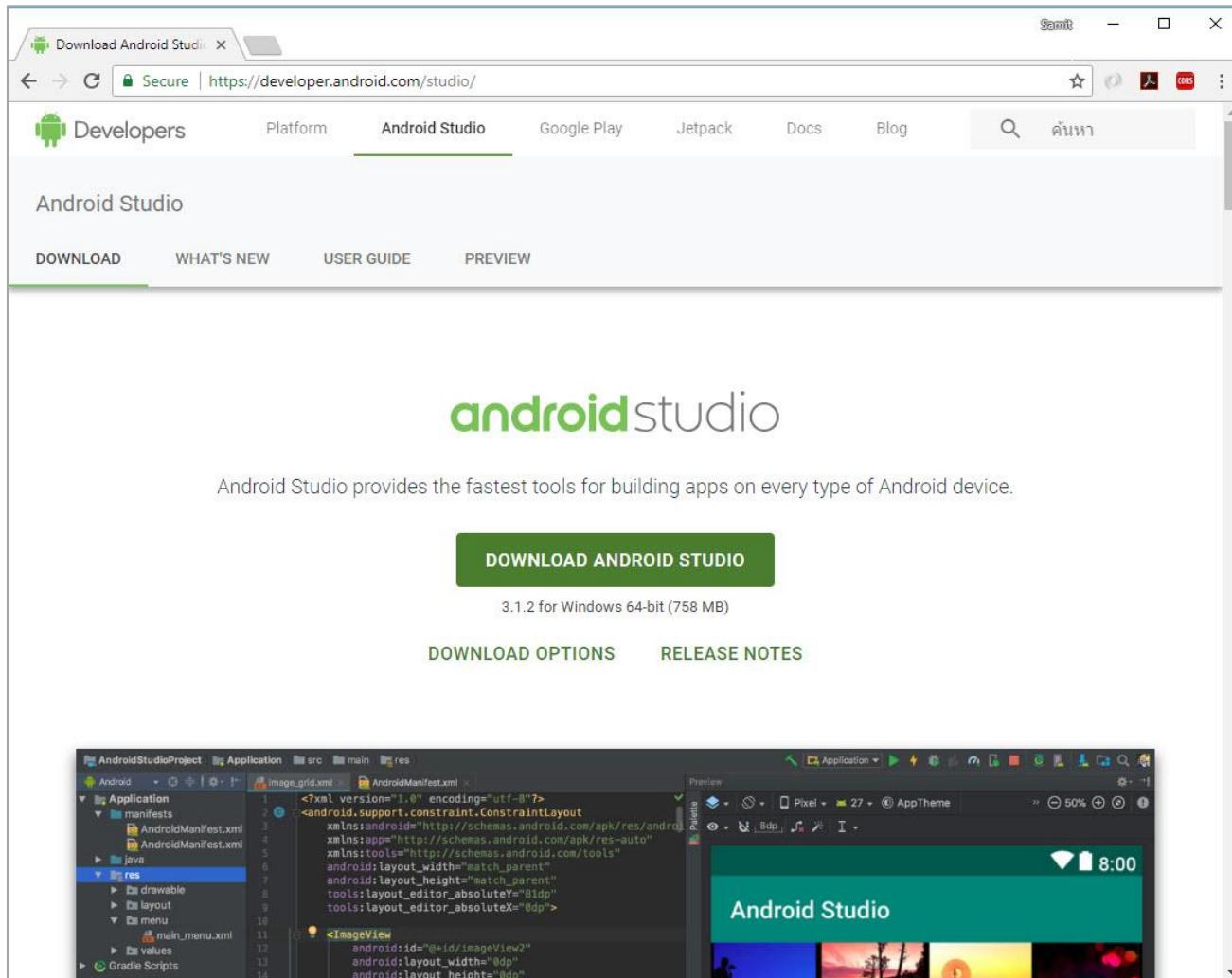
เลือก path ตามค่าเริ่มต้นได้เลย



รอติดตั้งจนแล้วเสร็จ คลิกปุ่ม Finish

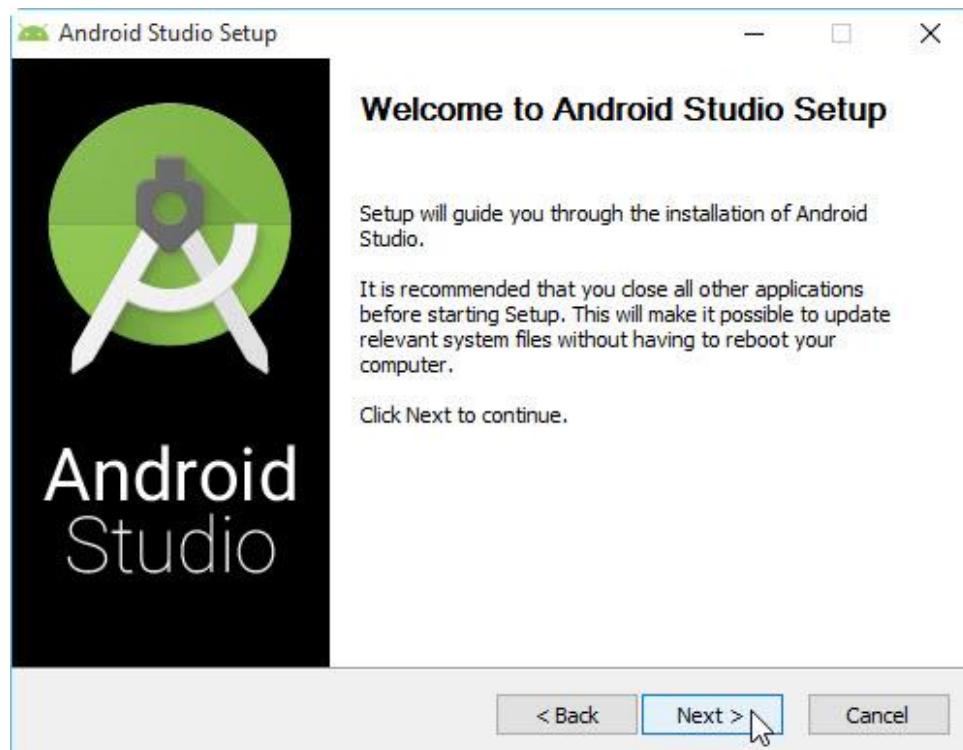
วิธีการติดตั้ง Android Studio และการ Config Path บน Windows และ Mac

Android Studio เป็นเครื่องมือในการพัฒนาแอปบน Android โดยมาพร้อมชุด SDK ทำให้เราไม่ต้องมาตั้งค่าอะไรเพิ่มเติมเอง ลงเสร็จพร้อมใช้งานในการพัฒนาแอป android ทันที ในการอบรมนี้เราจะใช้ android studio เพื่อโหลดชุด SDK มาใช้งานเท่านั้น เราไม่ได้เข้าไปสร้างแอปผ่านตัวเครื่องมือนี้แต่อย่างใด

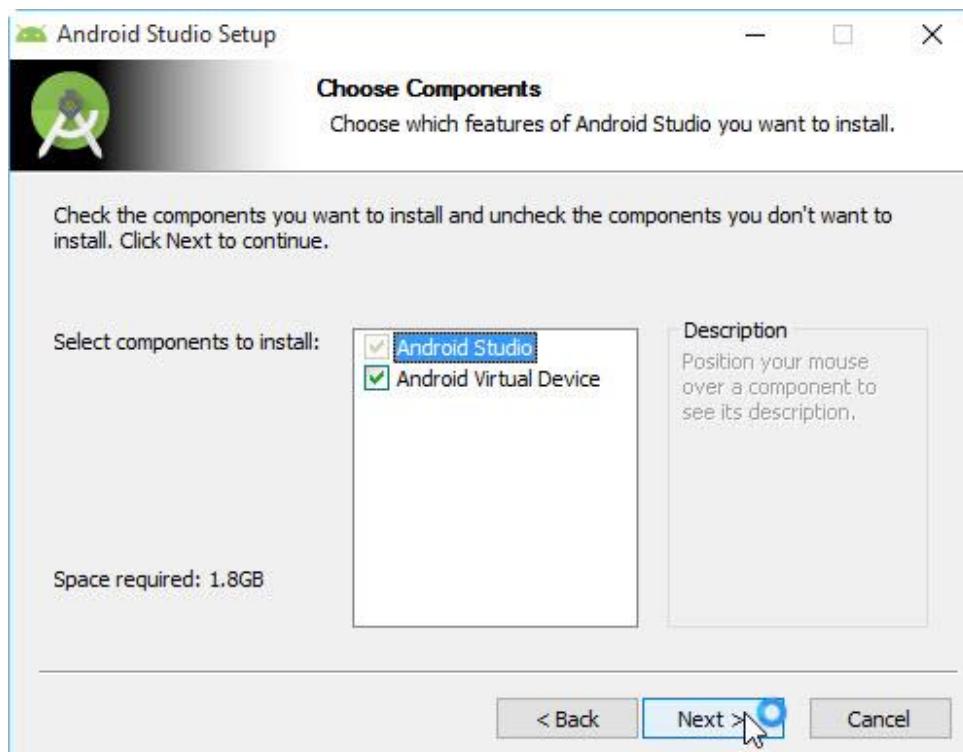


เข้าไปดาวน์โหลดติดตั้ง Android Studio ทั้งบน Mac และ Windows ได้ที่

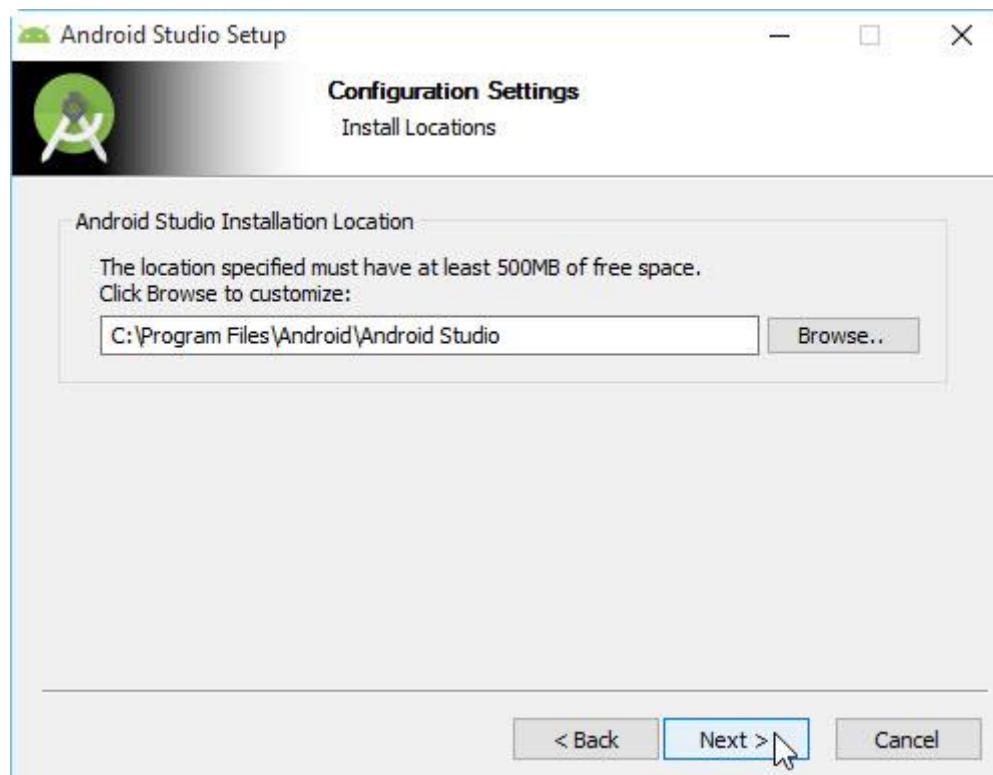
<https://developer.android.com/studio/> (ปัจจุบันที่เขียนเอกสารฉบับนี้เป็นเวอร์ชัน 3.1.2)



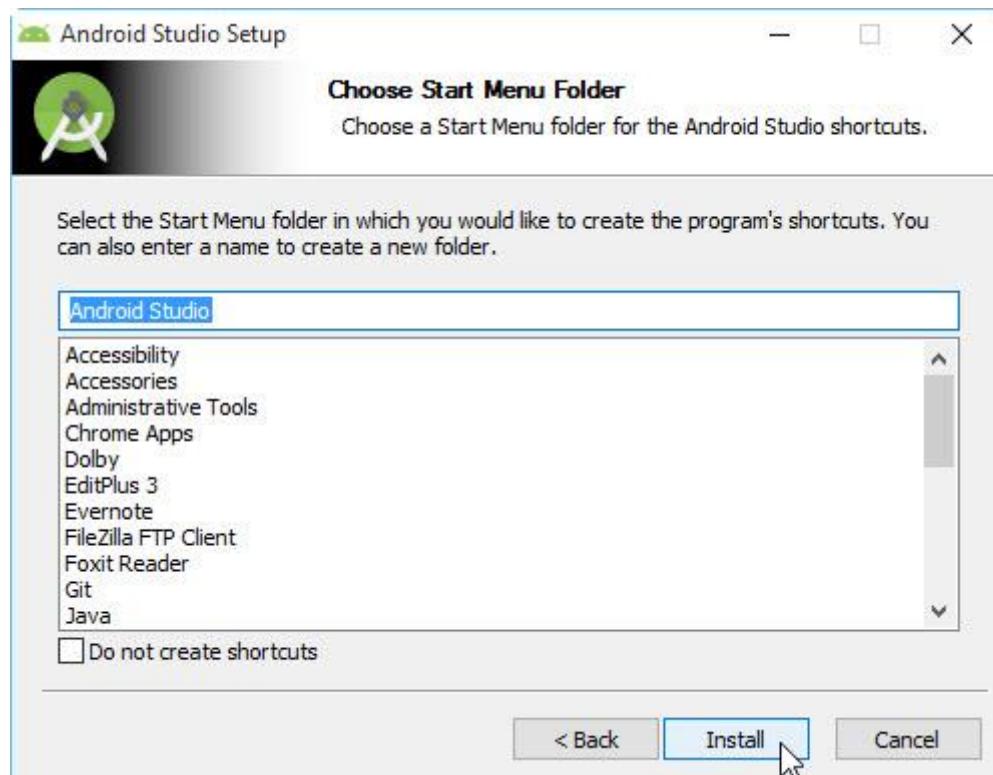
คลิก Next เริ่มทำการติดตั้ง



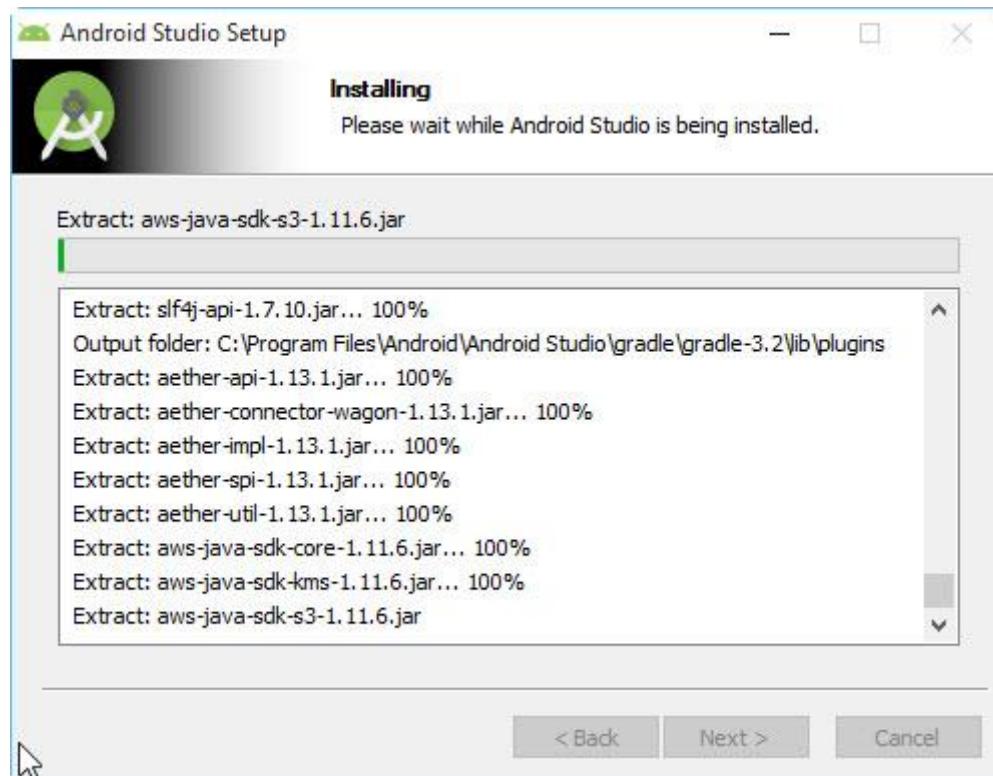
เลือกทั้ง Android Studio และ Android Virtual Device และคลิก Next



เลือก Path ที่ต้องการ (แนะนำตามค่าเริ่มต้น) และคลิก Next



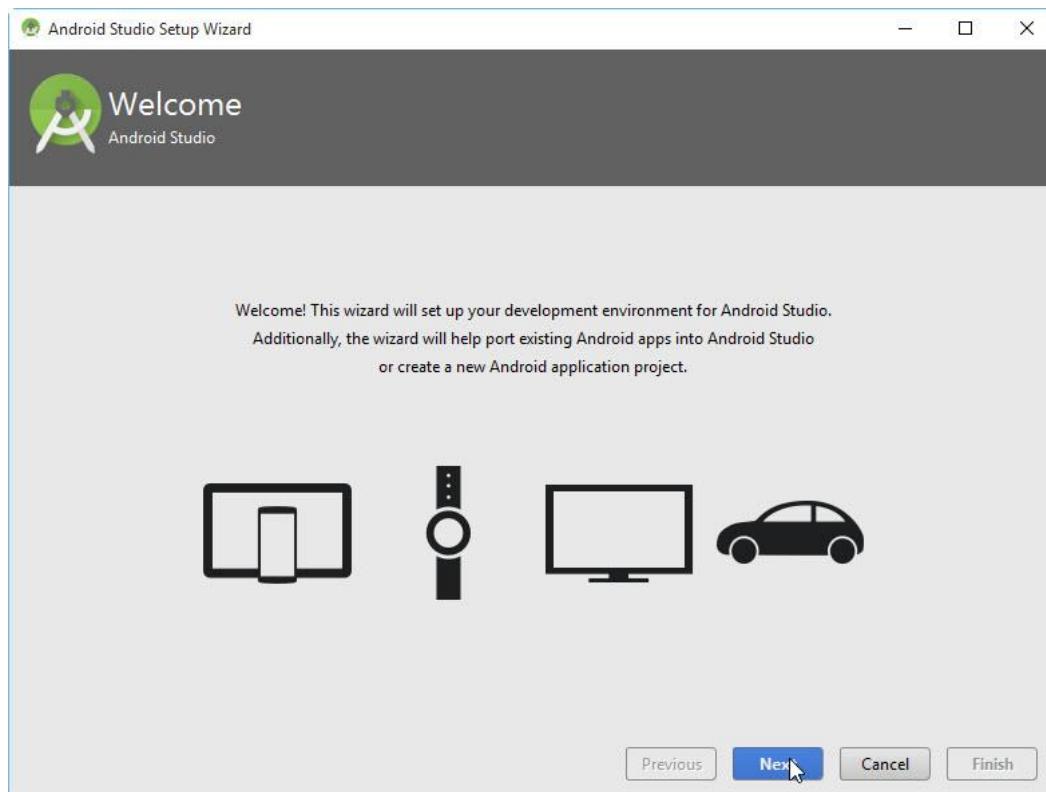
คลิก Install



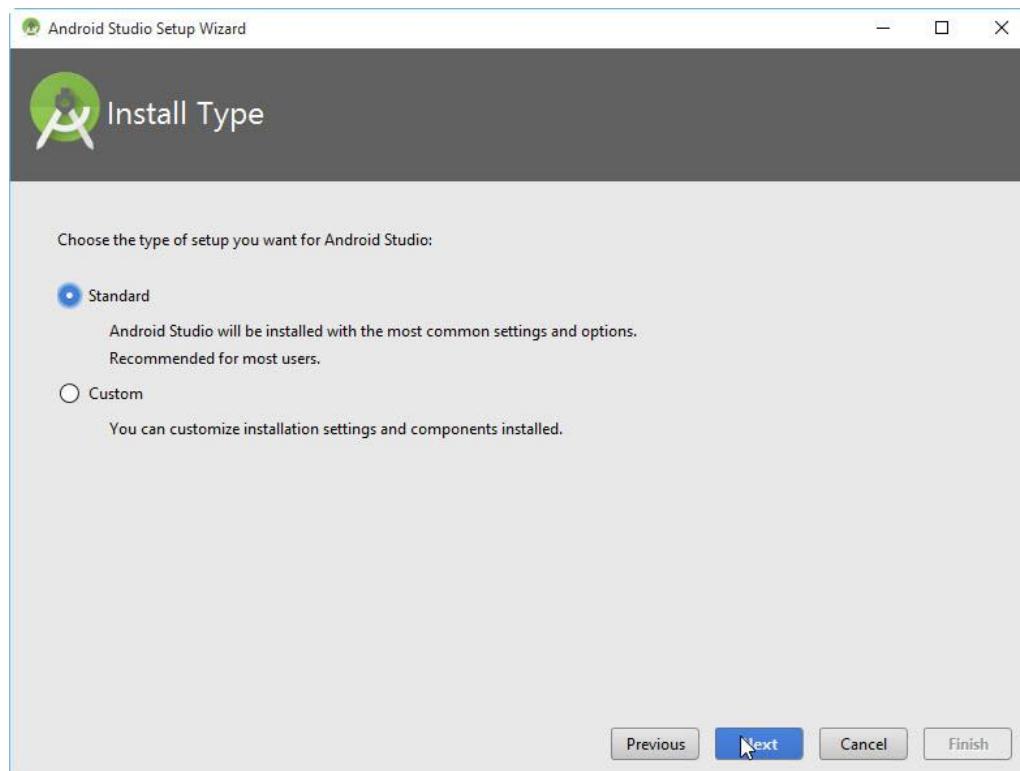
รอกระบวนการติดตั้ง จนแล้วเสร็จ



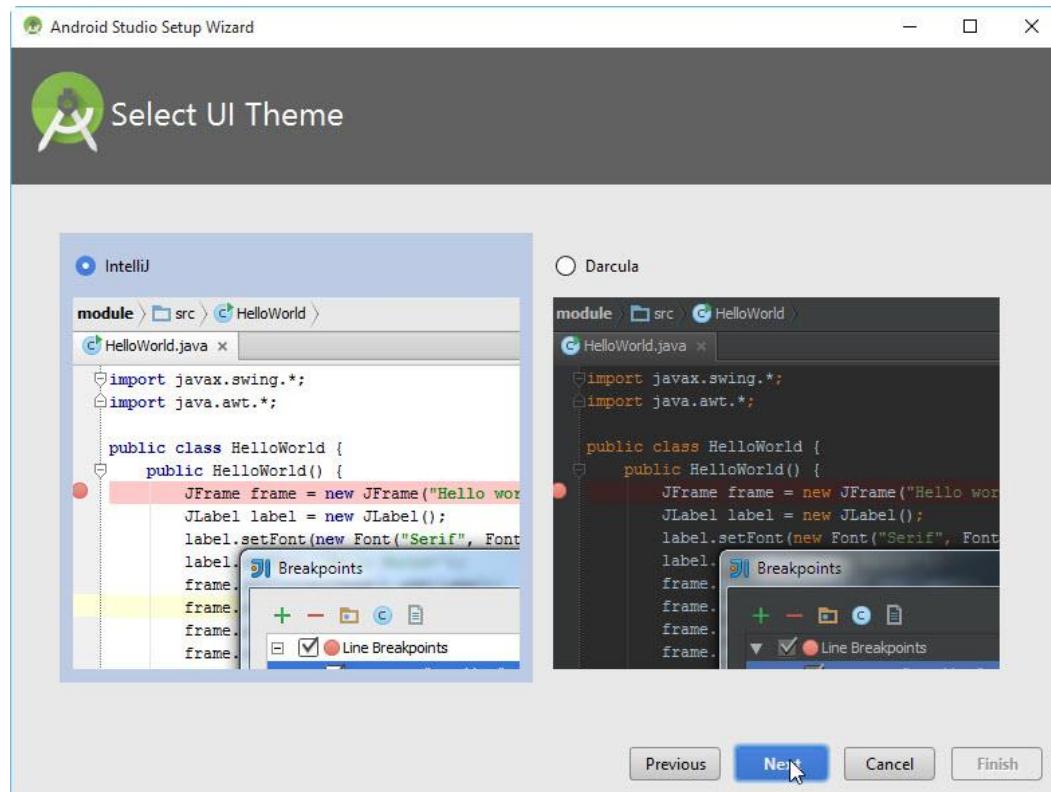
ติดตั้งเสร็จแล้วคลิกปุ่ม Finish



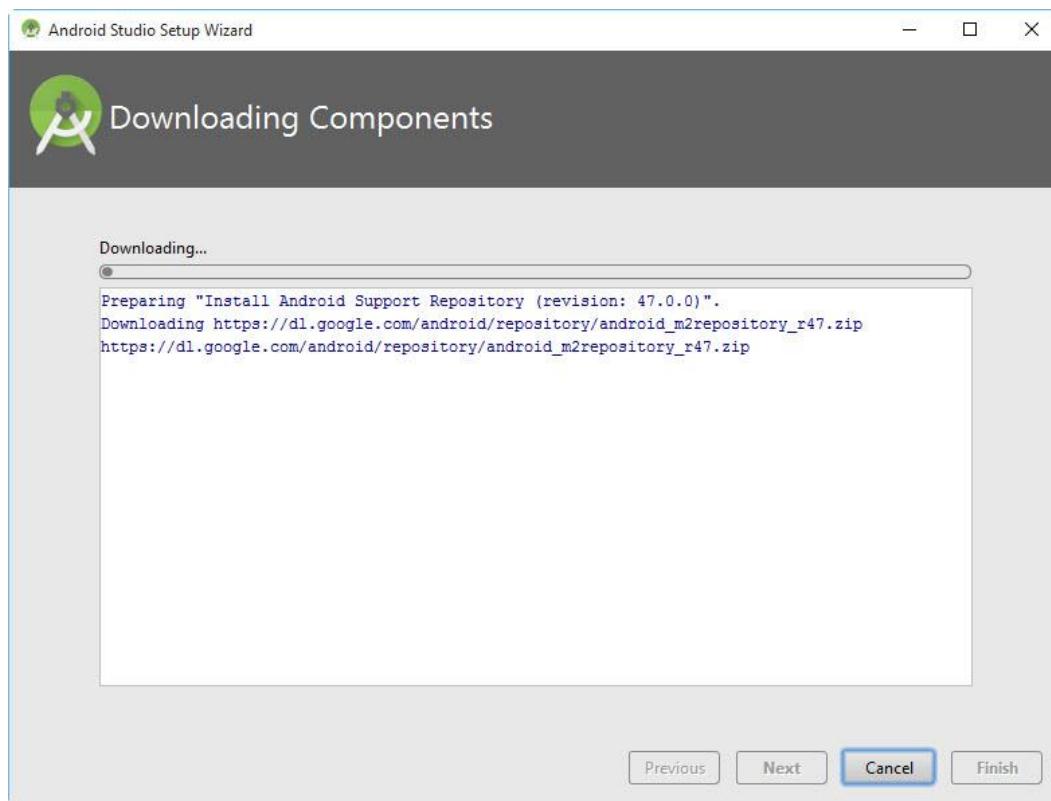
เมื่อติดตั้งเสร็จ เปิดโปรแกรม Android Studio ขึ้นมาจะพบหน้าต่างนี้ คลิกปุ่ม Next



เลือกรูปแบบ Standard ตามรูป คลิกปุ่ม Next

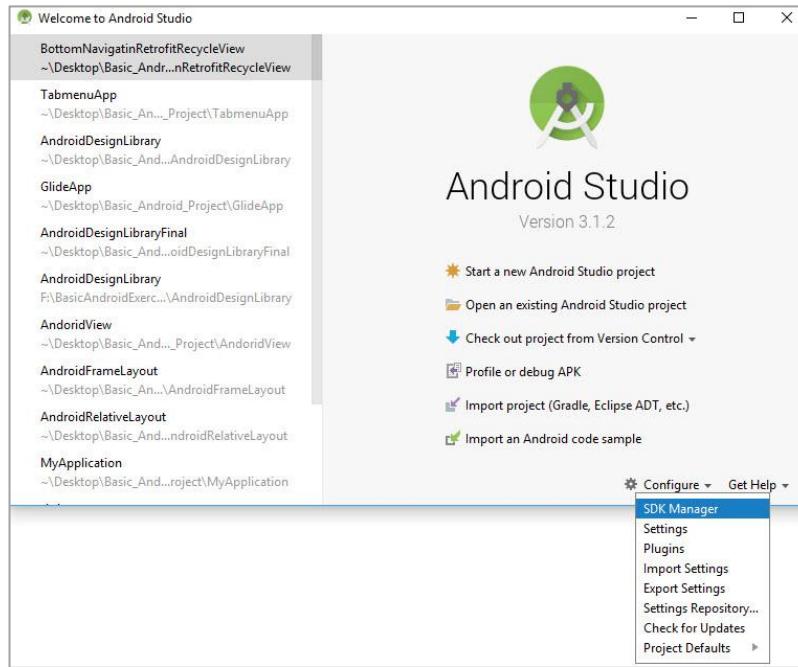


เลือก Theme ที่ต้องการ (แนะนำ Theme สีดำ Darcula)

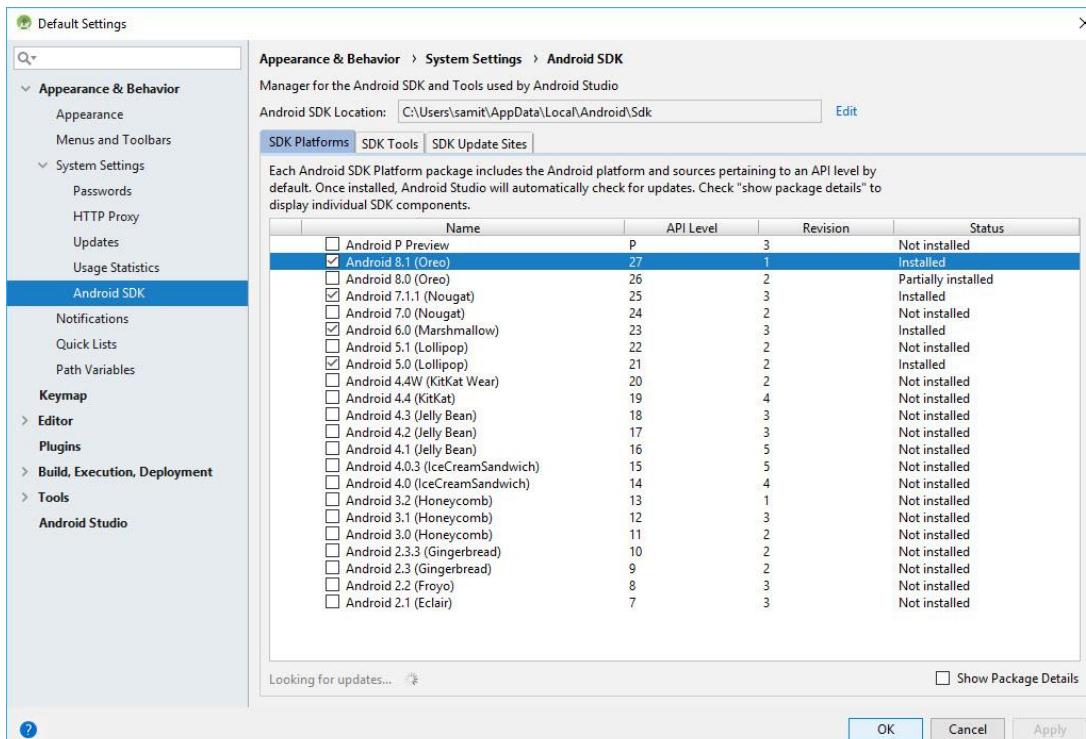


รอขั้นตอนการติดตั้ง Component จนแล้วเสร็จ

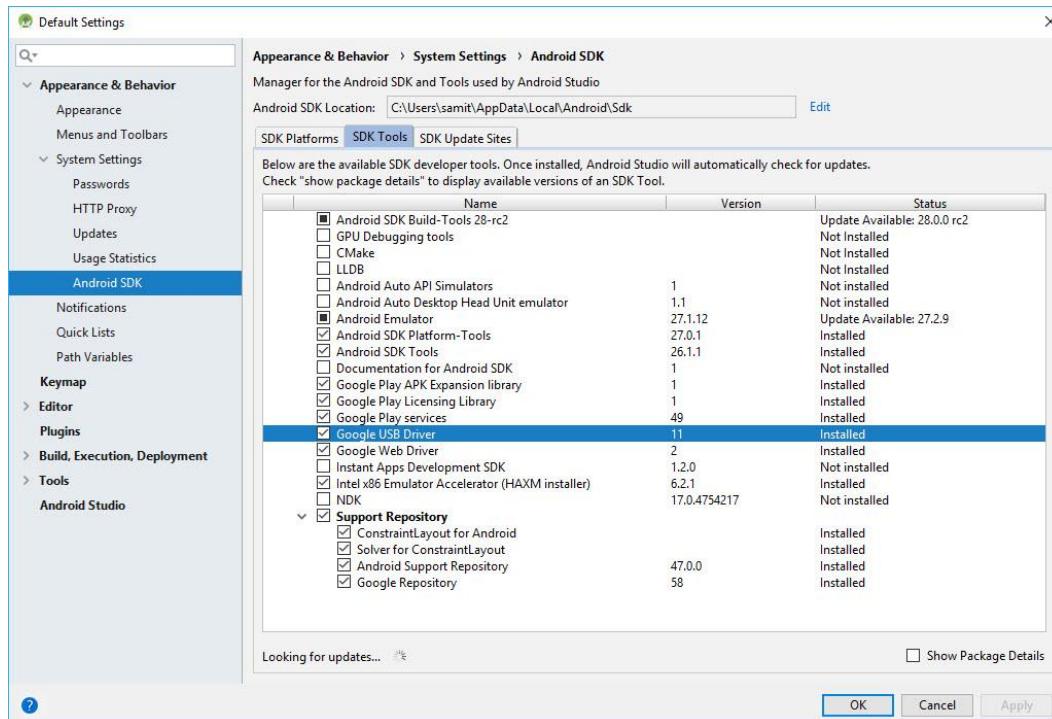
อัพเดตและติดตั้งชุด SDK เพิ่มเติม



หลังจากติดตั้ง Android Studio และอัพเดท component ต่าง ๆ แล้วเสร็จ จะพบหน้าจอ Welcome to Android Studio ดังภาพ ให้คลิกที่เมนู Configure > SDK Manager



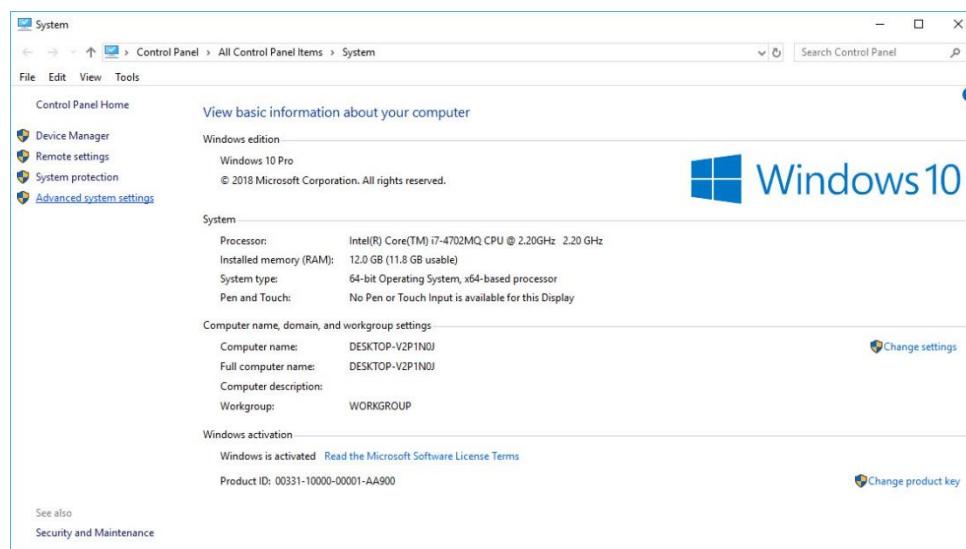
เลือกติดตั้งชุด SDK Platform ที่ต้องการ เช่น Android 8.1, Android 7.1.1 หรือ Android 6.0 ติดตั้งเพียงตัวเดียวเวอร์ชันล่าสุด (ไม่แนะนำ Preview version) ก็ได้



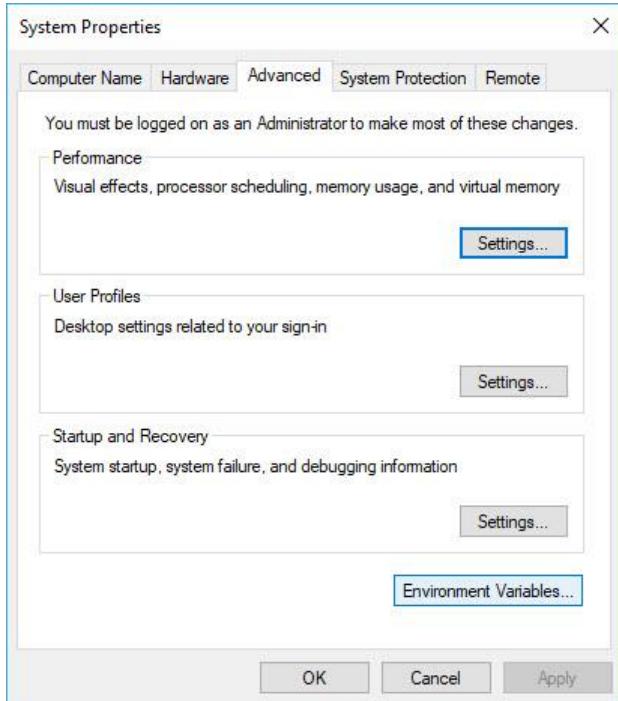
ที่ Tab SDK Tools เลือก Google USB Driver และ Google Play services ดังภาพ เสร็จแล้วคลิกปุ่ม OK และ รอการติดตั้งอัพเดทจนแล้วเสร็จ (จะใช้เวลาค่อนข้างนาน)

การกำหนด Android HOME ให้กับ Windows / Mac

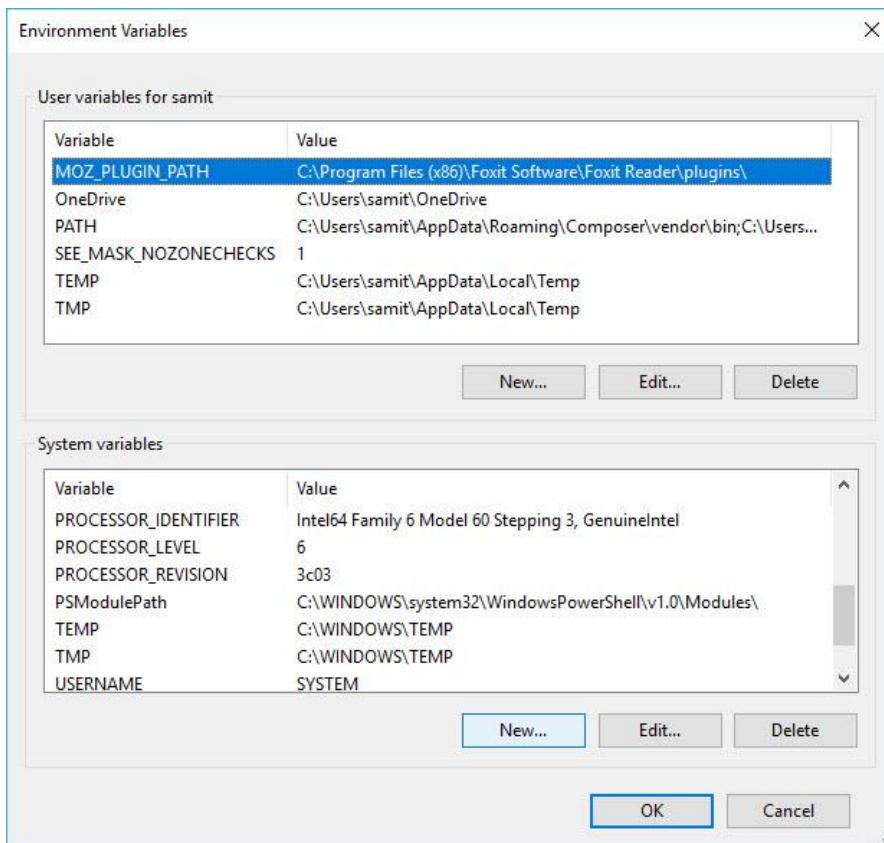
กำหนด Path Android Home บน Windows



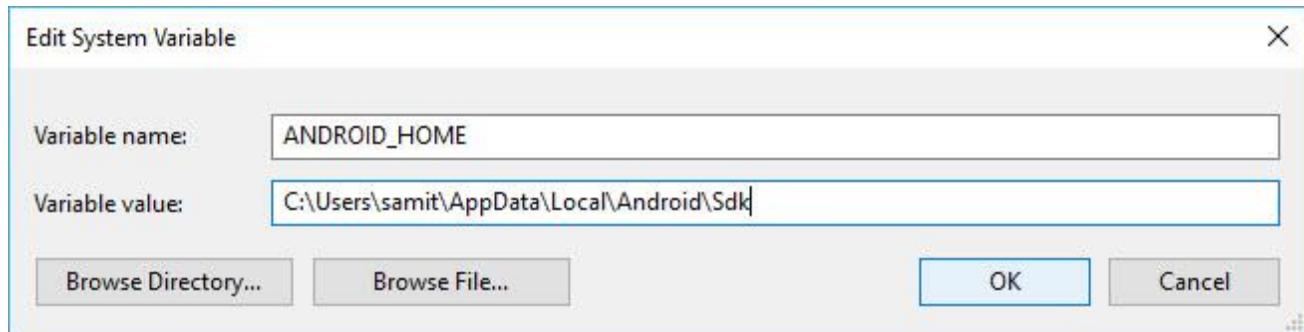
เข้าหน้า System ดังภาพ คลิกเลือกที่ Advanced system setting



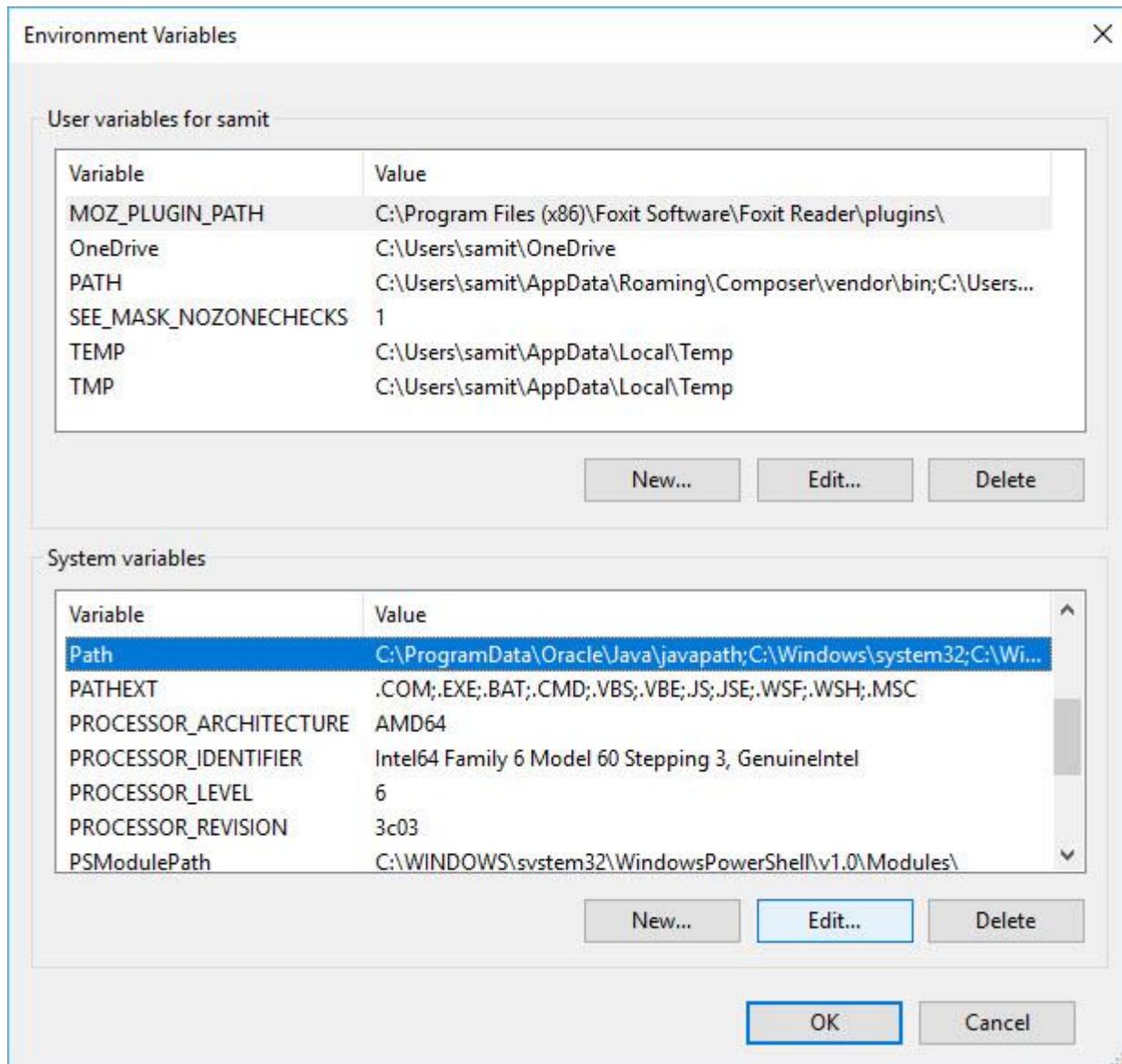
คลิกปุ่ม Environment Variables ดังภาพ



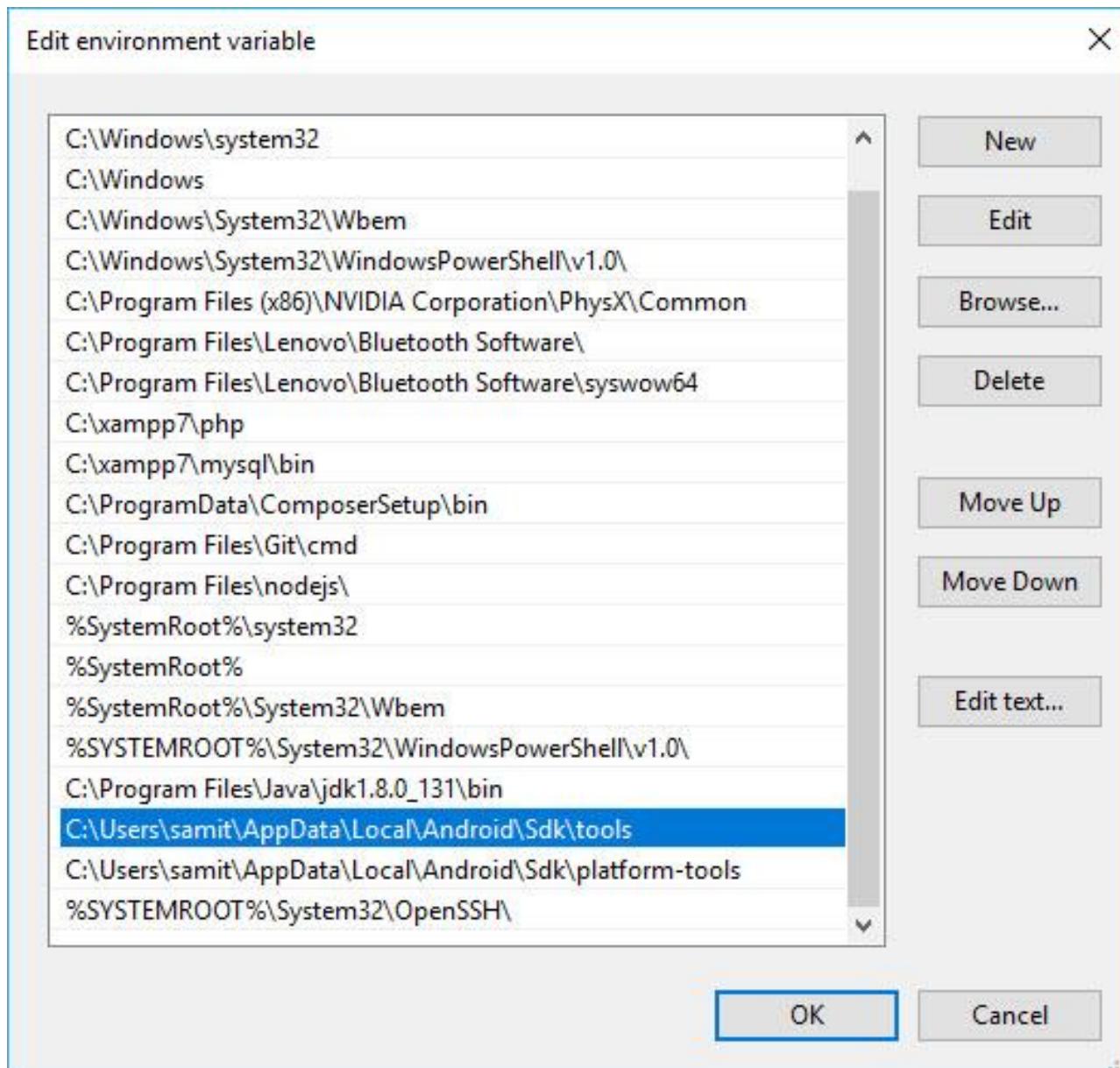
คลิกที่ปุ่ม New.. ด้านล่าง (ส่วนของ System variables)



ใส่ Variable name: เป็น ANDROID_HOME และ Variable value: เป็น
 C:\Users\ชื่อผู้ใช้\AppData\Local\Android\Sdk เรียบร้อยแล้วคลิกปุ่ม OK



จากนั้นคลิกเลือกที่ Path และกดปุ่ม Edit...

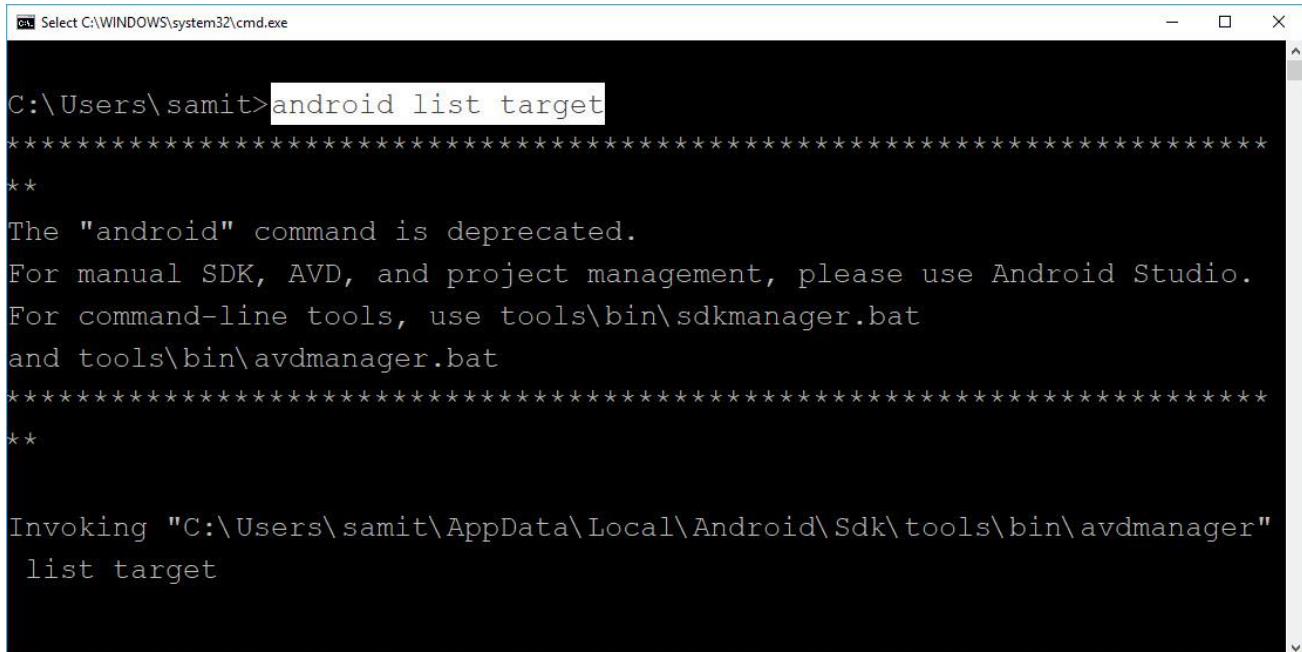


เพิ่ม Path โดยคลิกที่ปุ่ม New และทำการป้อน path

`C:\Users\samit\AppData\Local\Android\Sdk\tools` และ

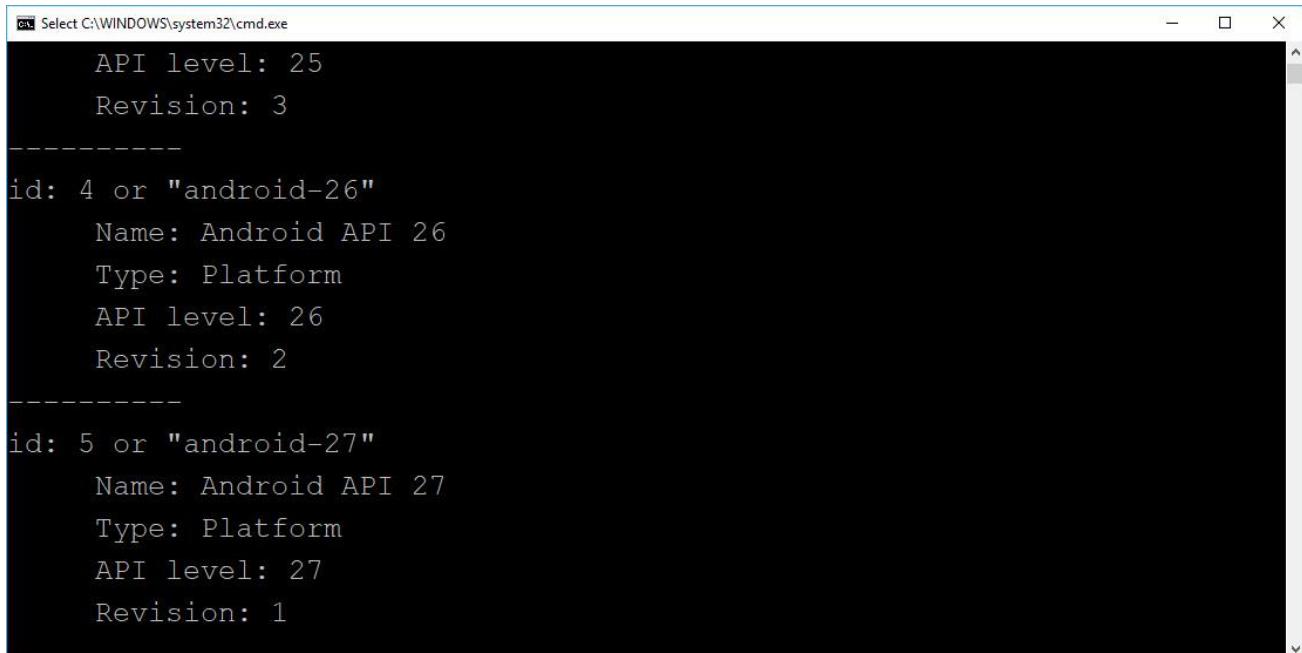
`C:\Users\samit\AppData\Local\Android\Sdk\platform-tools`

เรียบร้อยแล้วคลิกปุ่ม OK ออกไปจากทุกหน้าที่เปิดเข้ามา



```
C:\Users\samit>android list target
*****
** The "android" command is deprecated.
For manual SDK, AVD, and project management, please use Android Studio.
For command-line tools, use tools\bin\sdkmanager.bat
and tools\bin\avdmanager.bat
*****
** Invoking "C:\Users\samit\AppData\Local\Android\Sdk\tools\bin\avdmanager"
list target
```

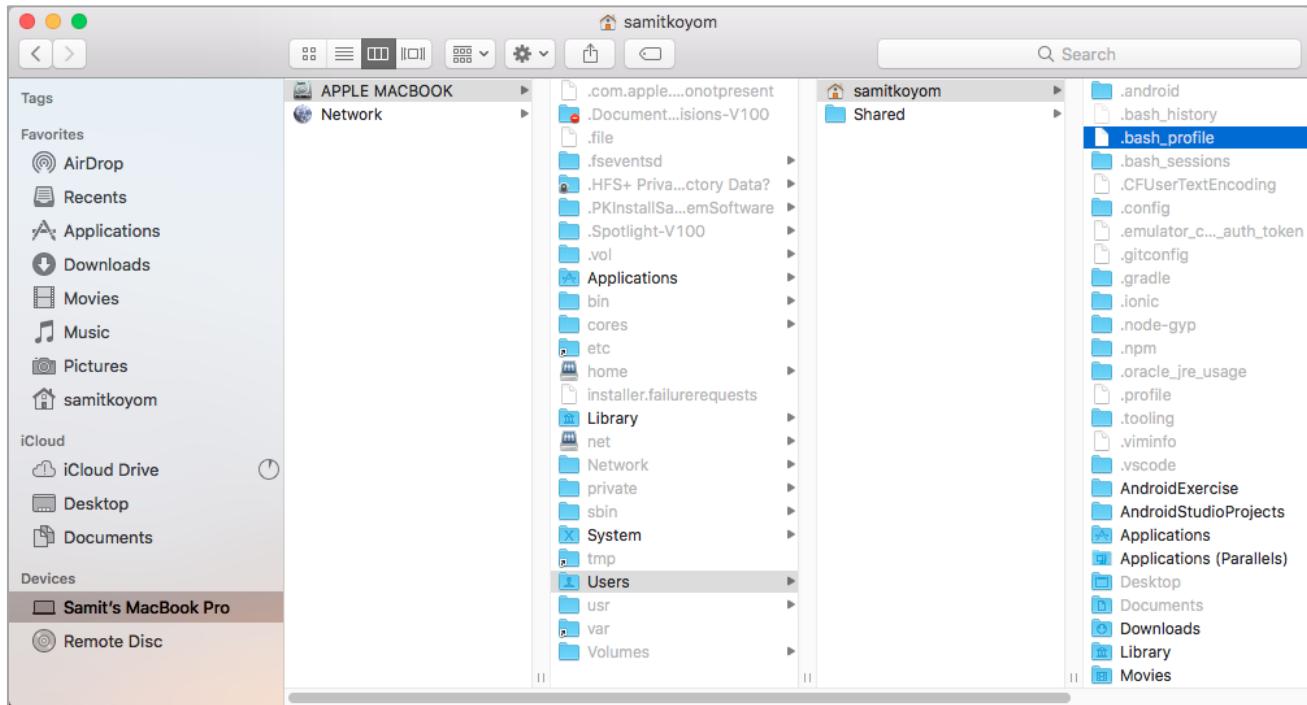
เปิด Command Prompt ขึ้นมาแล้วทำการพิมพ์คำสั่ง android list target เพื่อตรวจสอบความเรียบร้อย



```
API level: 25
Revision: 3
-----
id: 4 or "android-26"
    Name: Android API 26
    Type: Platform
    API level: 26
    Revision: 2
-----
id: 5 or "android-27"
    Name: Android API 27
    Type: Platform
    API level: 27
    Revision: 1
```

หากแสดงรายการ SDK ของ Android ขึ้นมาดังภาพก็ถือว่าเรียบร้อยดี

กำหนด Path Android Home บน Mac



เข้าไปสร้างไฟล์ .bash_profile ใน root user ที่คุณใช้งานอยู่

```
export JAVA_HOME=$(/usr/libexec/java_home)
export PATH="/Users/samitkoyom/Library/Android/sdk/platform-tools":$PATH
export PATH="/Users/samitkoyom/Library/Android/sdk/tools":$PATH
```

เพิ่มคำสั่ง

```
export PATH="/Users/your_user/Library/Android/sdk/platform-tools":$PATH
export PATH="/Users/your_user/Library/Android/sdk/tools":$PATH
```

ทำการบันทึกไฟล์และปิดไป

```

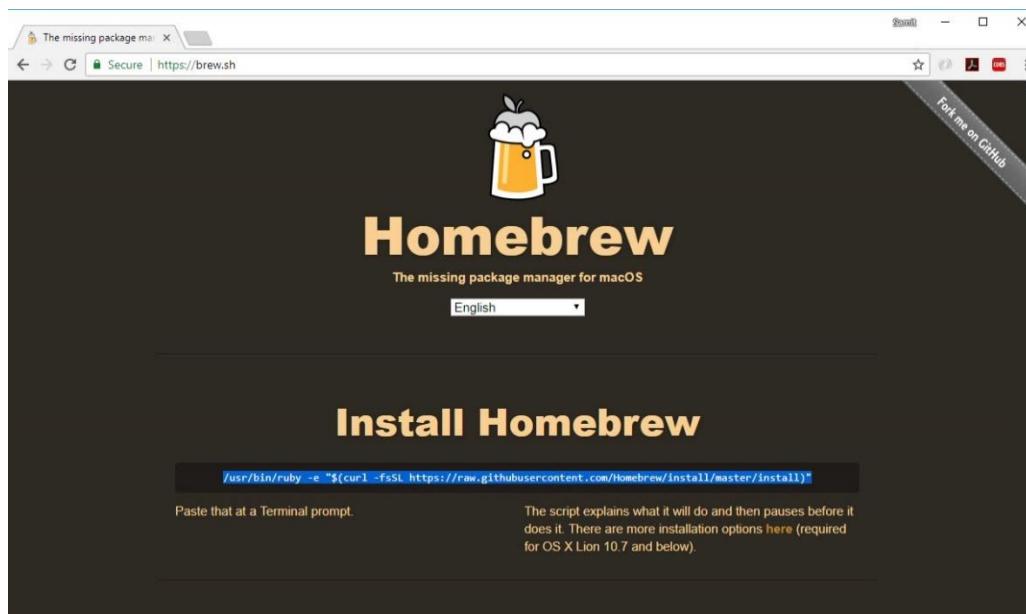
MacbookPro:~ samitkoyom$ android list target
*****
The "android" command is deprecated.
For manual SDK, AVD, and project management, please use Android Studio.
For command-line tools, use tools/bin/sdkmanager and tools/bin/avdmanager
*****
Running /Users/samitkoyom/Library/Android/sdk/tools/bin/avdmanager list target

Available Android targets:=====] 100% Fetch remote repository...
-----
id: 1 or "android-25"
    Name: Android API 25
    Type: Platform
    API level: 25
    Revision: 3
-----
id: 2 or "android-26"
    Name: Android API 26
    Type: Platform
    API level: 26
    Revision: 2
-----
id: 3 or "android-27"
    Name: Android API 27

```

เปิด Terminal พิมพ์คำสั่ง `android list target` เพื่อทำการตรวจสอบ จะแสดงรายการ SDK ของ Android ที่ติดตั้งไว้ดังภาพ

การติดตั้ง Home brew บน mac เพื่อเรียกใช้งาน Gradle สำหรับทดสอบ Run Android



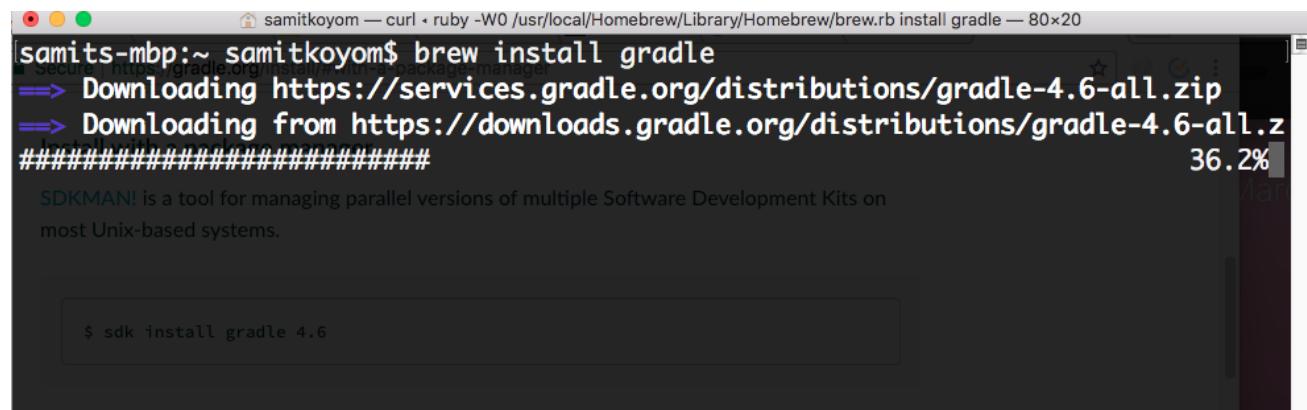
เข้าไปคัดลอกคำสั่งติดตั้งที่เว็บไซต์ <https://brew.sh/>

เปิด Terminal ขึ้นมาแล้วป้อนคำสั่ง

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

รอ....ติดตั้งจนแล้วเสร็จ

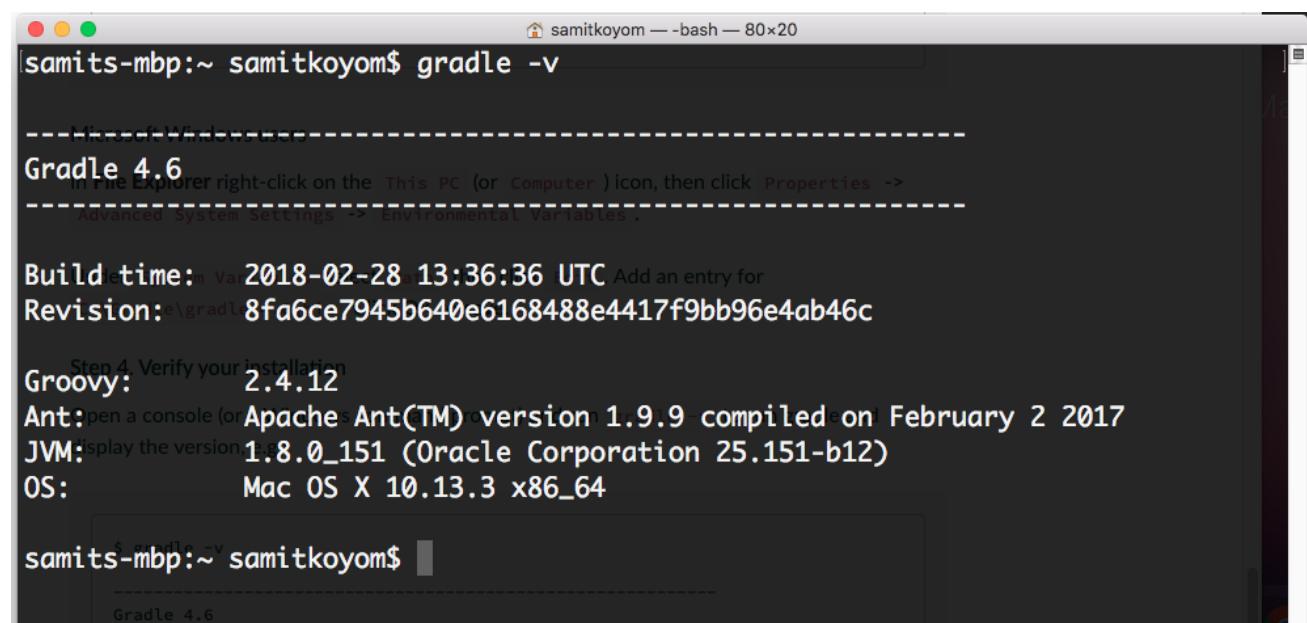
จากนั้นทำการติดตั้ง Gradle ด้วยคำสั่งด้านล่างนี้



```
samitkoyom — curl + ruby -W0 /usr/local/Homebrew/Library/Homebrew/brew.rb install gradle — 80x20
samits-mbp:~ samitkoyom$ brew install gradle
==> Downloading https://services.gradle.org/distributions/gradle-4.6-all.zip
==> Downloading from https://downloads.gradle.org/distributions/gradle-4.6-all.z
#####
SDKMAN! is a tool for managing parallel versions of multiple Software Development Kits on
most Unix-based systems.

$ sdk install gradle 4.6
```

brew install gradle รอ...ติดตั้งจนแล้วเสร็จ

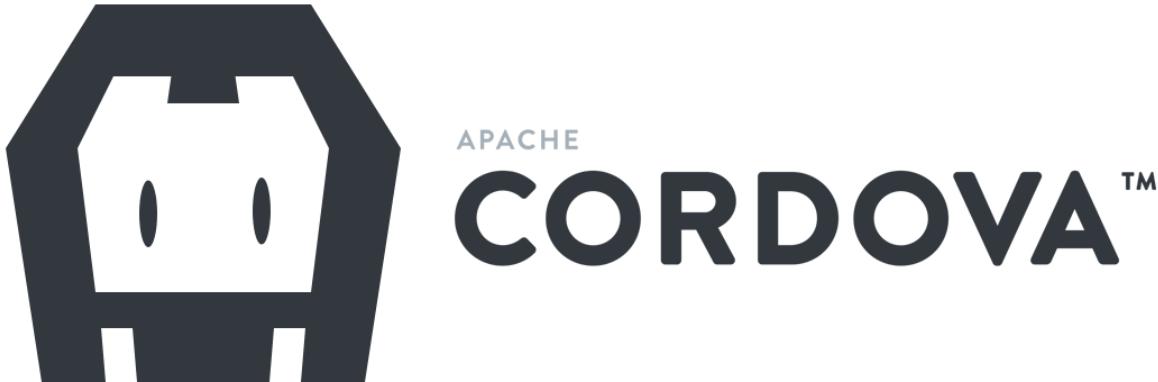


```
samits-mbp:~ samitkoyom$ gradle -v
-----
Gradle 4.6
-----[in the Explorer right-click on the This PC (or Computer) icon, then click Properties ->
----- Advanced System Settings -> Environmental Variables .]
-----[in the Variables section, click New .]
-----[in the Variable Name field, type 'GRADLE_HOME' and click OK .]
-----[in the Variable Value field, type 'C:\gradle\gradle-4.6' and click OK .]
-----[click OK to close the Properties dialog .]
-----[click OK to close the Advanced System Settings dialog .]
-----[click OK to close the System dialog .]
-----[click OK to close the Control Panel dialog .]
-----Build time: 2018-02-28 13:36:36 UTC Add an entry for
-----Revision: 8fa6ce7945b640e6168488e4417f9bb96e4ab46c
-----Step 4. Verify your installation
-----Groovy: 2.4.12
-----Ant: Apache Ant(TM) version 1.9.9 compiled on February 2 2017
-----JVM: 1.8.0_151 (Oracle Corporation 25.151-b12)
-----OS: Mac OS X 10.13.3 x86_64
samits-mbp:~ samitkoyom$
```

พิมพ์คำสั่ง gradle -v เช็คเวอร์ชันของ Gradle หากพบเลขเวอร์ชันแสดงออกมาก็เป็นอันเรียบร้อย

การติดตั้ง Cordova และ Ionic framework บน Windows และ Mac

มาถึงขั้นตอนสำคัญสำหรับการเริ่มเขียนแอปแบบ Cross Platform ในการอบรมครั้งนี้ เราจะใช้ Ionic framework ซึ่งทำงานอยู่ภายใต้ Cordova ดังนั้นเราจะทำการติดตั้ง Cordova เสียก่อนดังนี้



คำสั่งติดตั้ง Cordova Framework บน Windows และ Mac

- บน Windows เปิด Command Prompt ขึ้นมาพิมพ์คำสั่ง `npm install -g cordova`
- บน Mac เปิด Terminal ขึ้นมาพิมพ์คำสั่ง `sudo npm install -g cordova`
- เมื่อติดตั้งเสร็จแล้ว พิมพ์คำสั่ง `cordova -v` ใน Command Prompt หรือ Terminal เพื่อตรวจสอบ

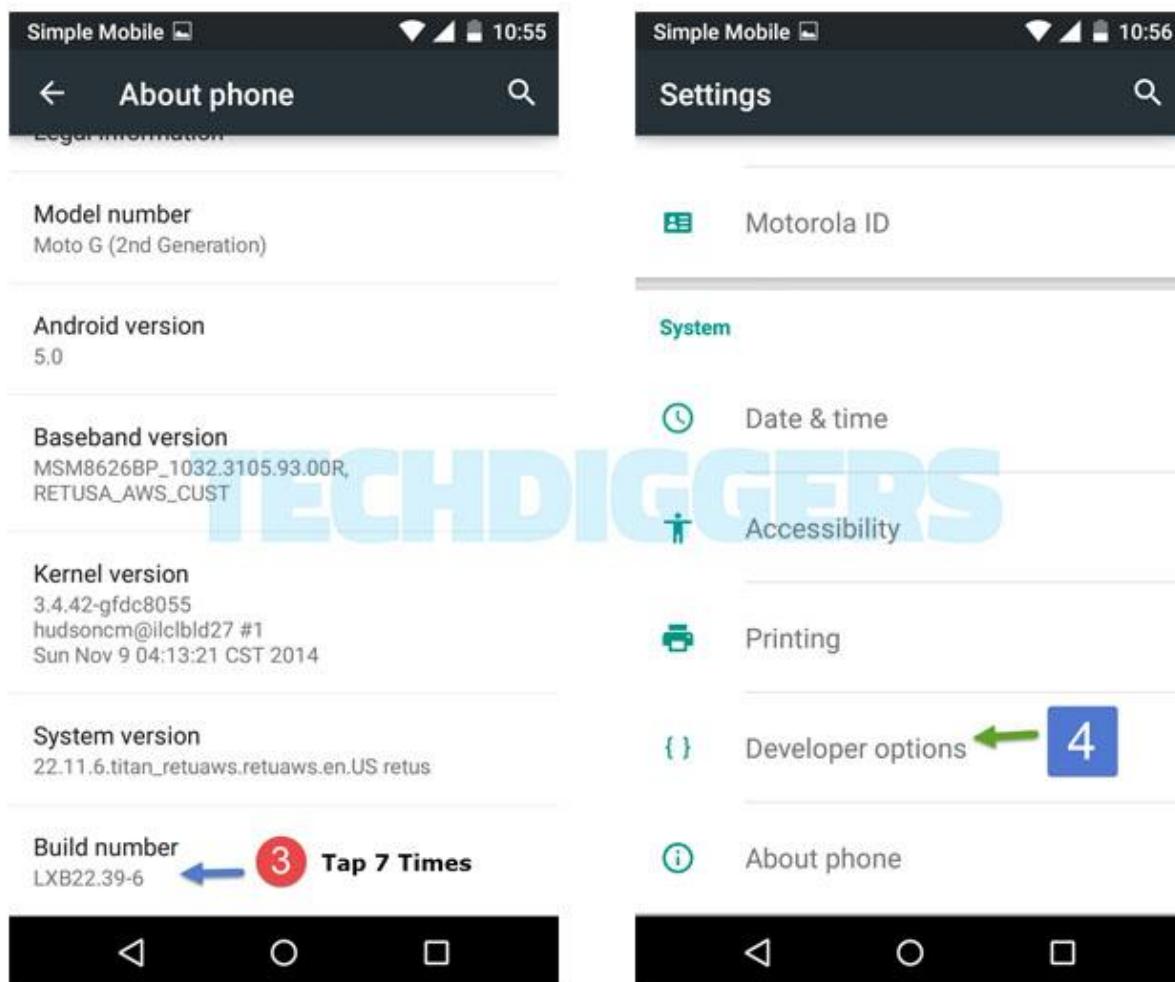


คำสั่งติดตั้ง Ionic Framework บน Windows และ Mac

- บน Windows เปิด Command Prompt ขึ้นมาพิมพ์คำสั่ง `npm install -g ionic`
- บน Mac เปิด Terminal ขึ้นมาพิมพ์คำสั่ง `sudo npm install -g ionic`
- เมื่อติดตั้งเสร็จแล้ว พิมพ์คำสั่ง `ionic -v` ใน Command Prompt หรือ Terminal เพื่อตรวจสอบ

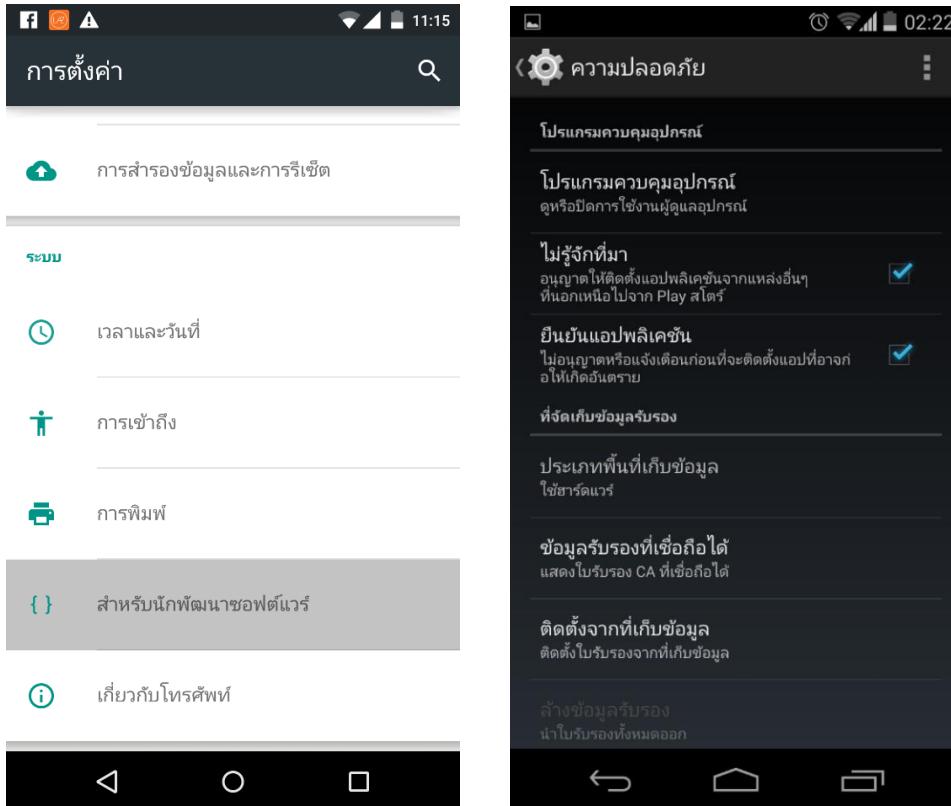
ลง Driver ADB interface บน Windows สำหรับเชื่อมต่อ กับ Smart Phone Android

เปิดใช้งาน Developer mode บนอุปกรณ์ Android

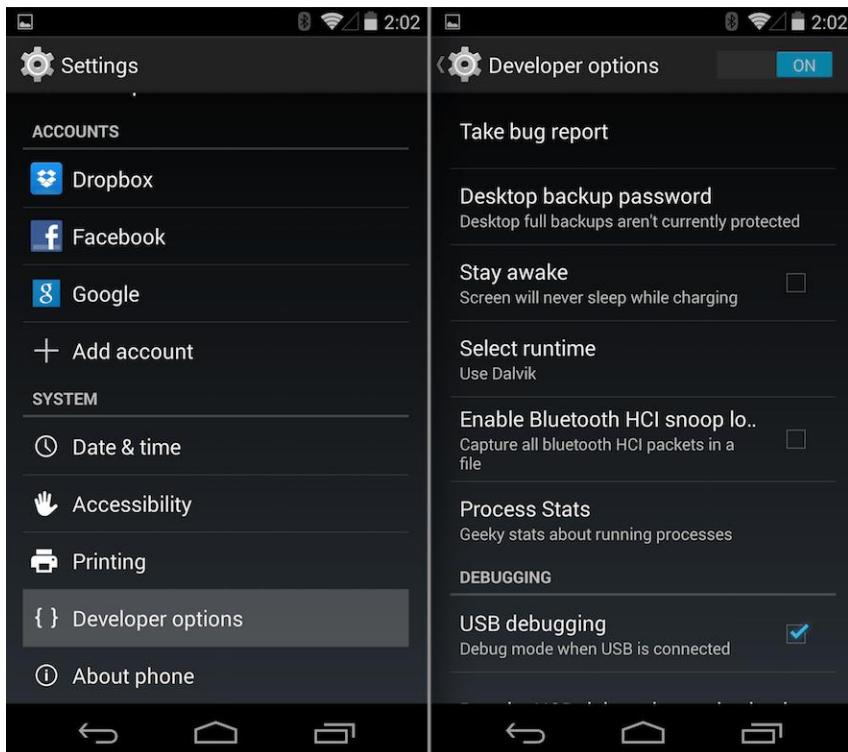


เข้าเมนู Setting ของมือถือ android มองหาเมนู About Phone (เกี่ยวกับโทรศัพท์) หาเมนู Build number (หมายเลขบิลด์) ทำการกดลงไป 7 ครั้ง จะเป็นการเปิดโหมด Developer Option ขึ้นมาดังภาพ

อนุญาติให้ติดตั้งแอพแอนดรอยด์ จาก Unknown Resource ได้

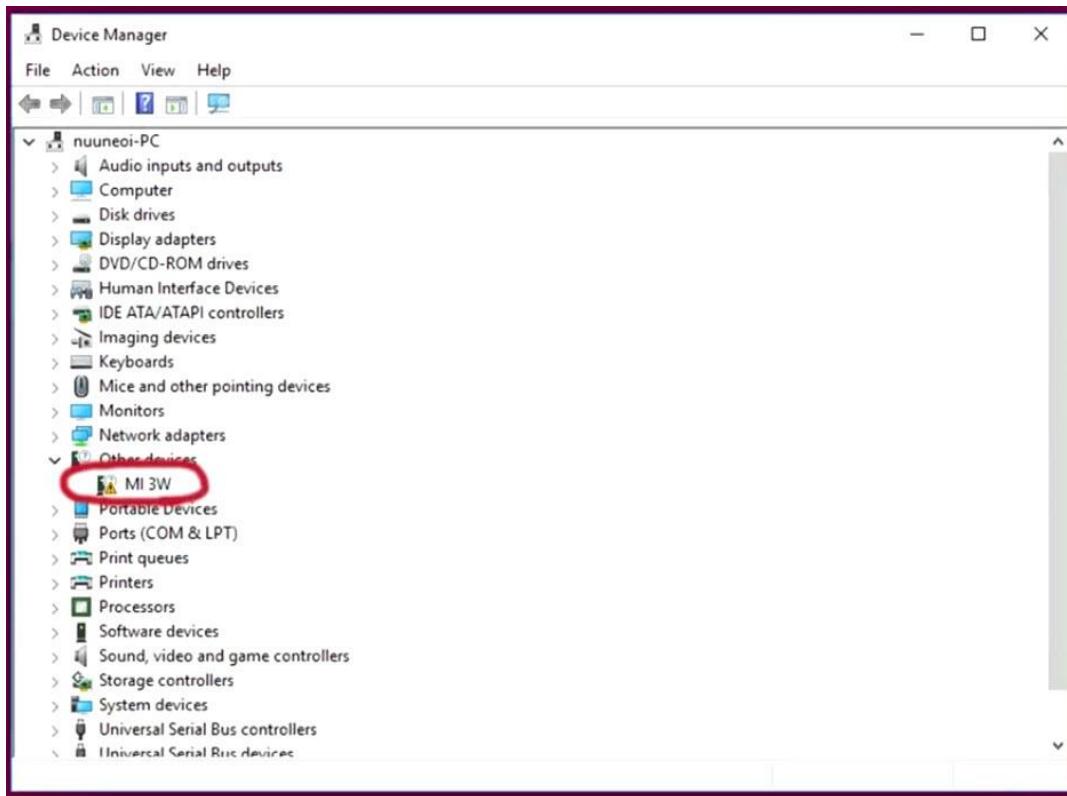


เปิด USB Debugging (การแก้ไขข้อบกพร่องของ USB)

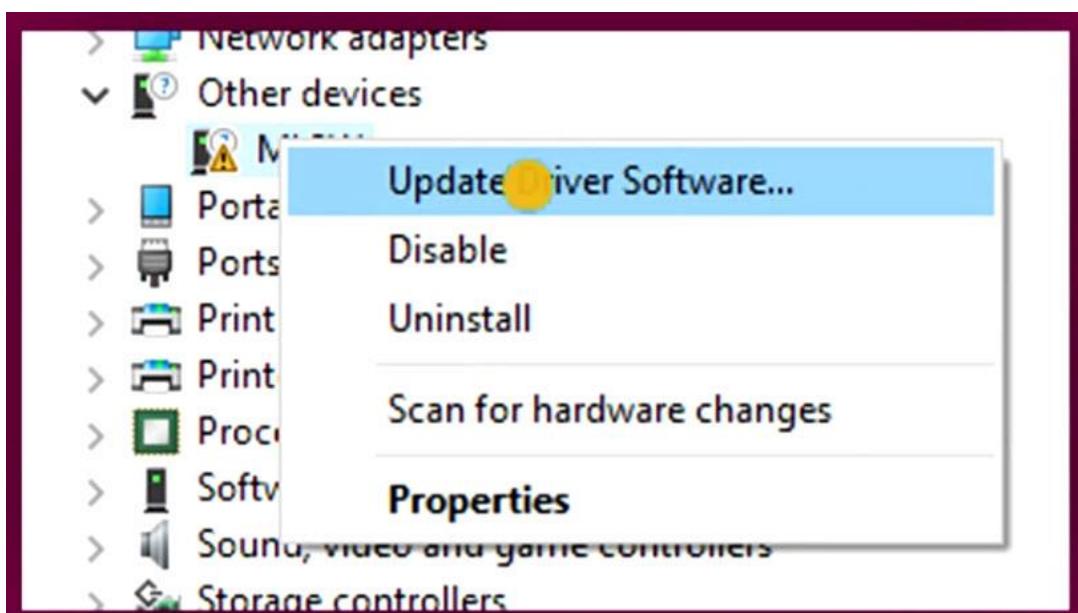


การติดตั้ง Driver ADB ให้กับมือถือแอนดรอยด์

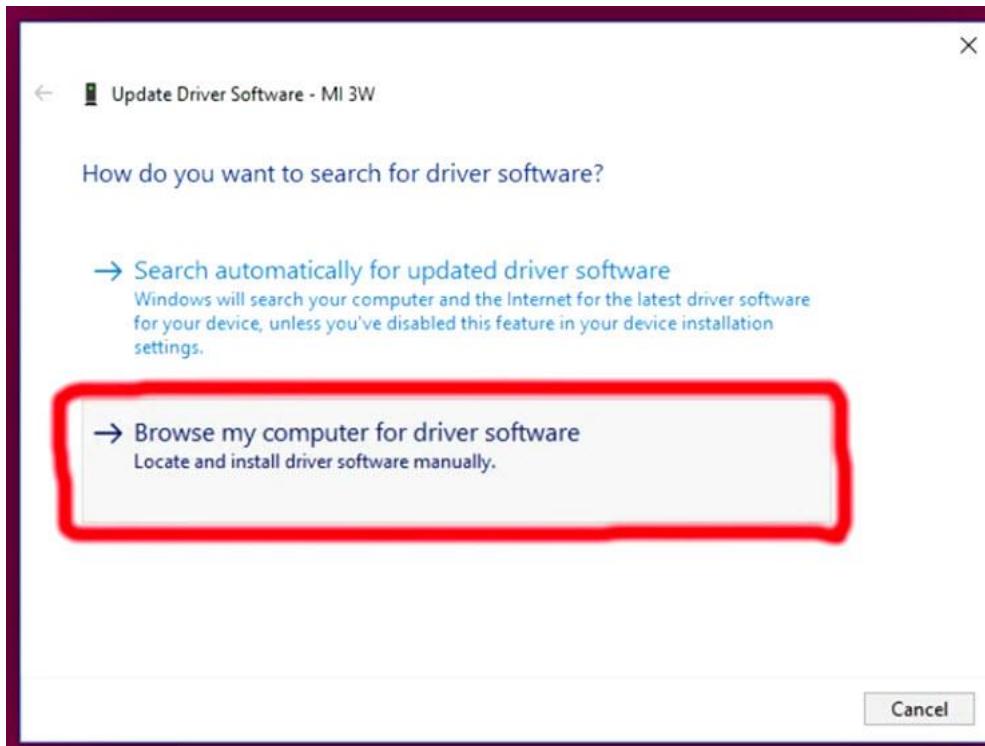
เชื่อมต่อสาย USB ระหว่างมือถือ Android กับคอมพิวเตอร์ที่ใช้แพ็ฒนาโปรแกรม



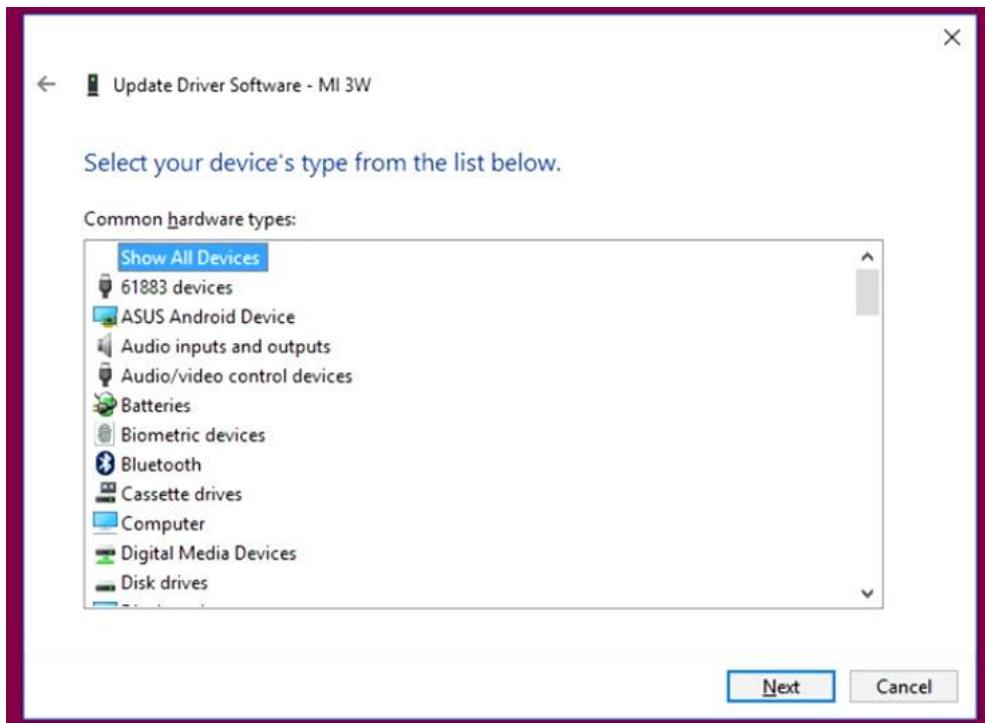
เข้าไปยัง Device Manager ของ Windows มองหาอุปกรณ์ที่เชื่อมต่อไว้



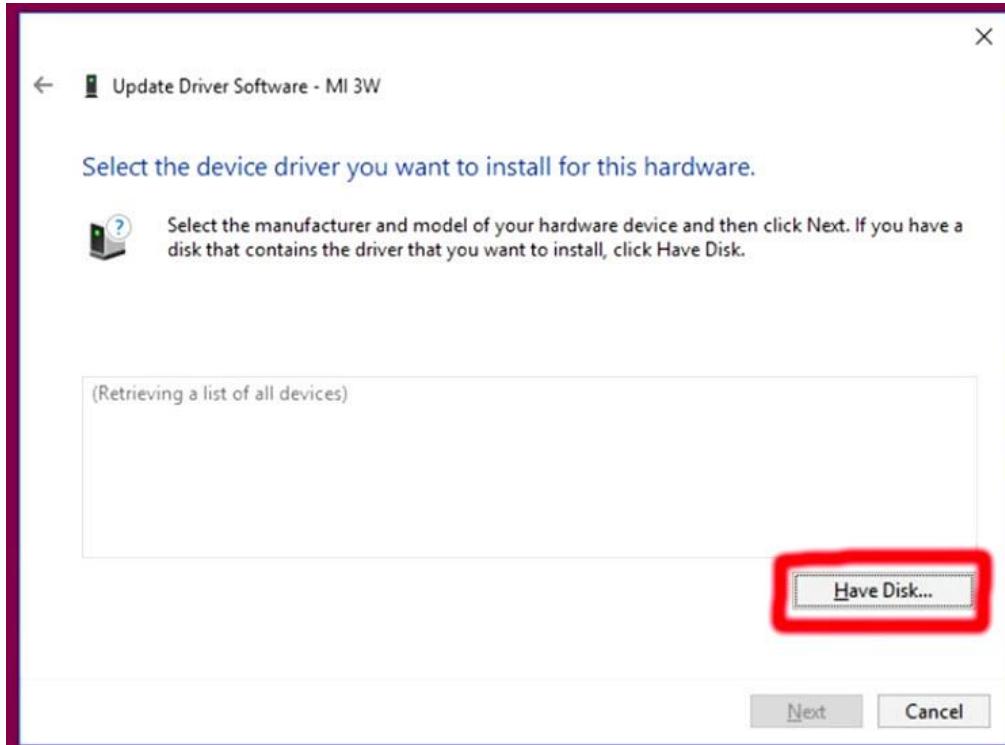
เลือก Update Driver Software...



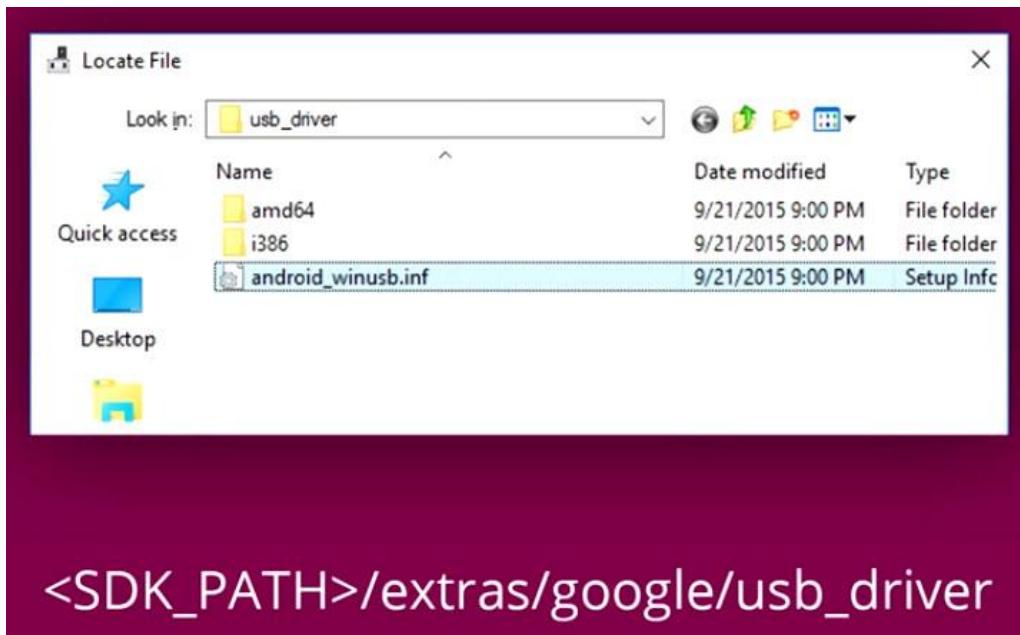
เลือก Browse my computer for driver software



เลือก Show All Devices และกดปุ่ม Next

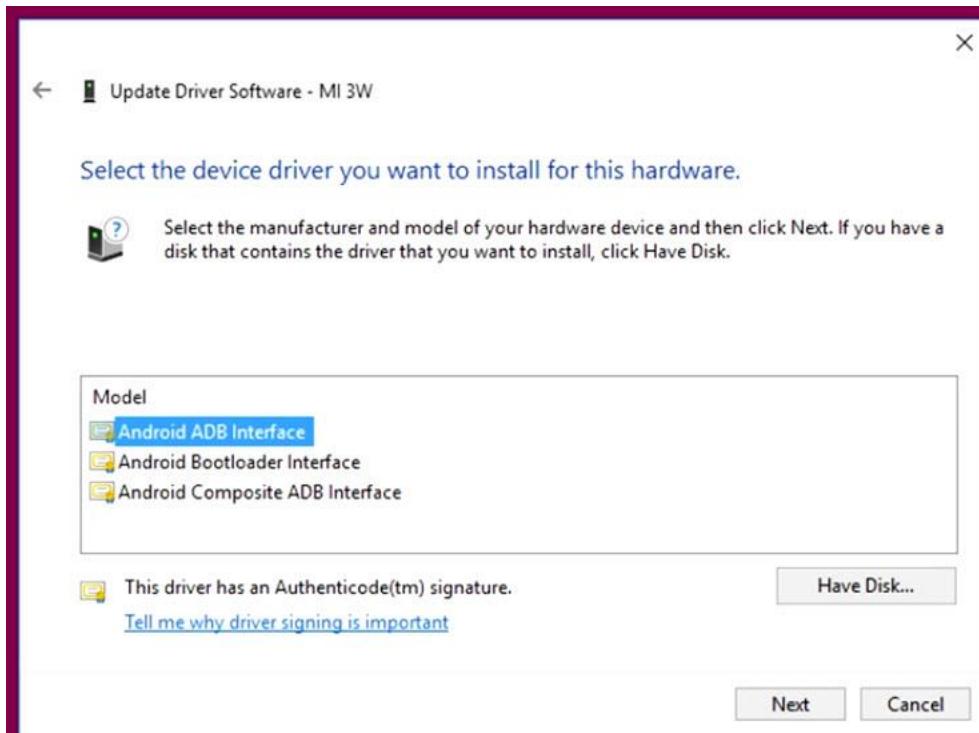


คลิกปุ่ม Have Disk.. ดังภาพ

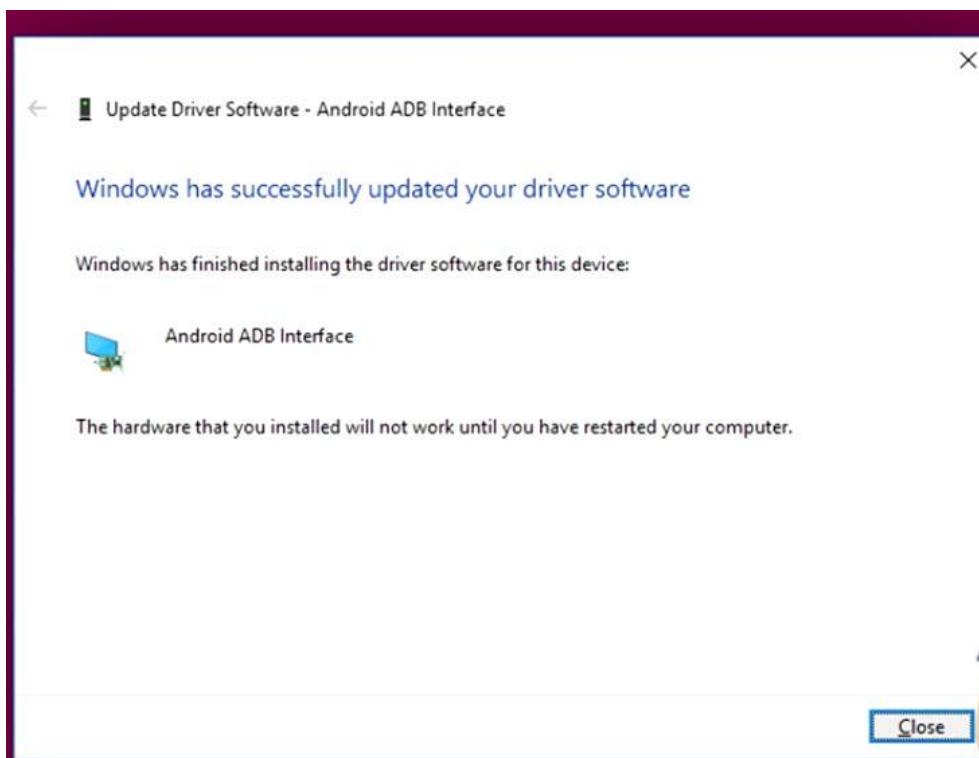


เลือก Driver จาก Path

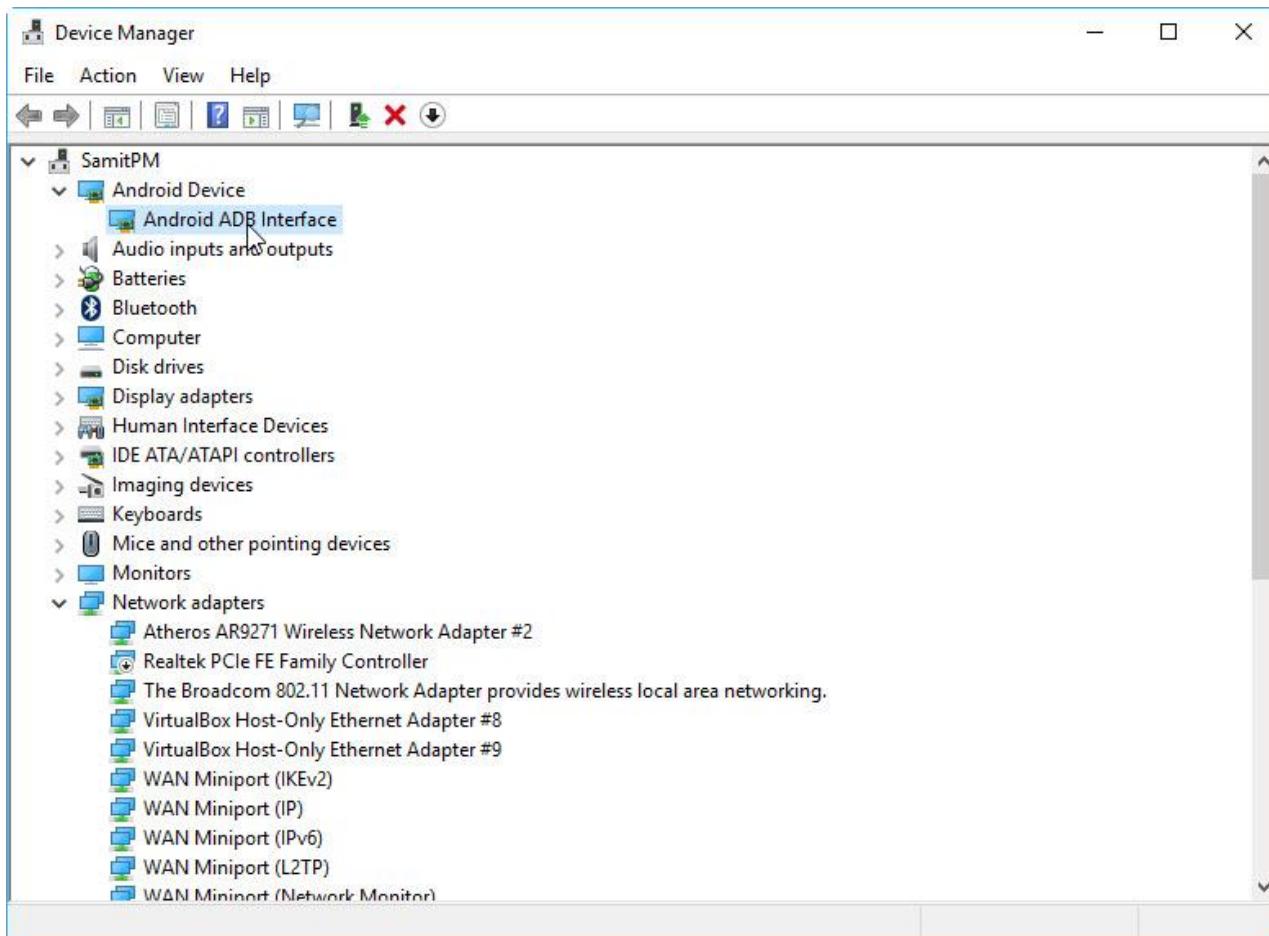
C:\Users\samit\AppData\Local\Android\Sdk\extras\google\usb_driver\android_winusb.inf



เลือก Android ADB Interface ดังภาพ



อัพเดท Driver เรียบร้อยคลิกปุ่ม Close ได้เลย



หากถูกต้องเรียบร้อยจะแสดงราย Android Device เป็น Android ADB Interface ดังภาพ

```
C:\Users\samit>adb devices
List of devices attached
* daemon not running; starting now at tcp:5037
* daemon started successfully
ENU7N15B05000171    device

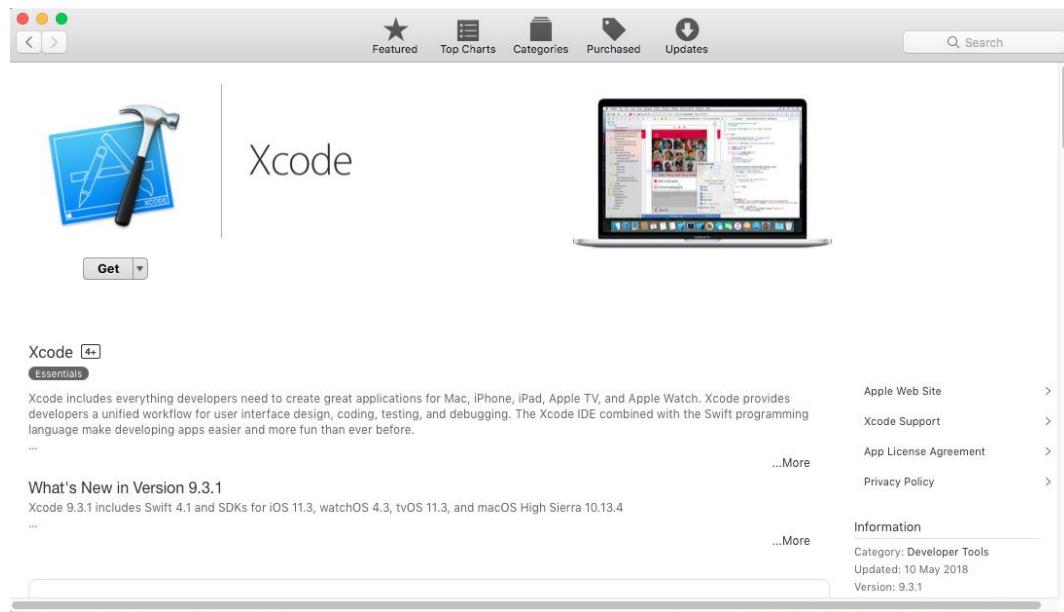
C:\Users\samit>
```

The screenshot shows a Command Prompt window with the path 'C:\Users\samit>'. The user has run the command 'adb devices'. The output shows the daemon was not running, so it started successfully, and one device is connected, identified by the serial number 'ENU7N15B05000171' and labeled as a 'device'.

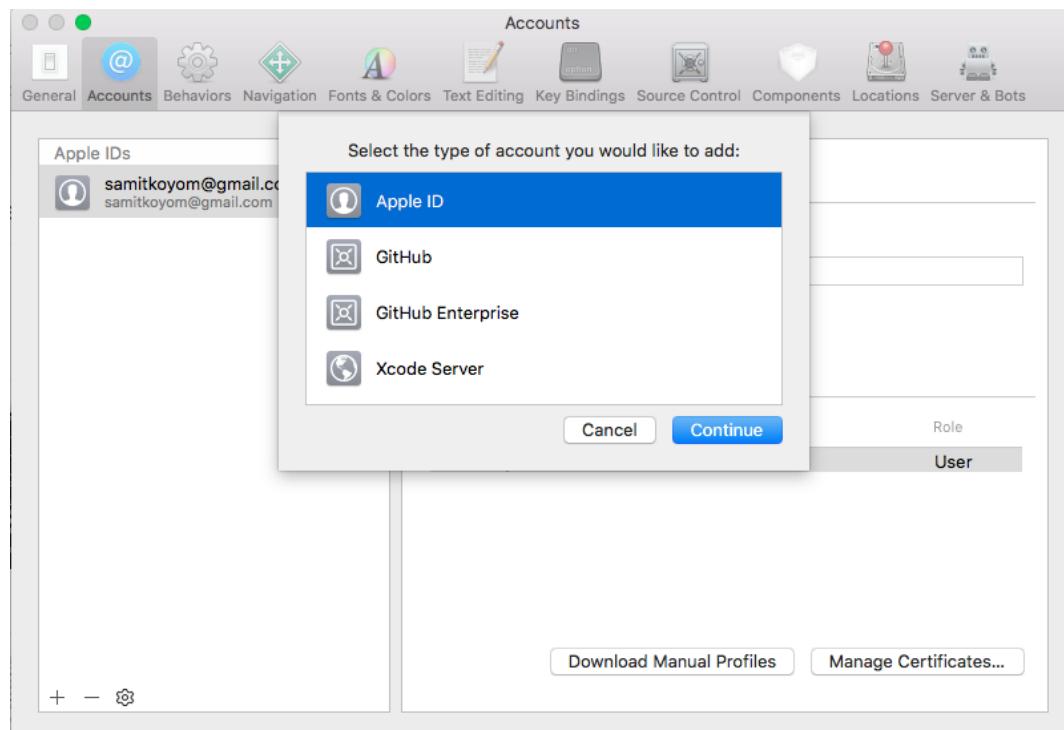
เปิด Command Prompt หรือ Terminal บน mac พิมพ์คำสั่ง adb devices จะแสดงรายการอุปกรณ์อยู่ในเครื่องเรา ที่เราได้เชื่อมต่ออยู่ ทั้งนี้เราสามารถใช้คำสั่ง adb shell ในการควบคุมและรันแอปพลิเคชันบน设备 ที่เราเชื่อมต่อ

เพิ่มเติมเครื่องมือสำหรับ Platform iOS (บนเครื่อง Mac เท่านั้น)

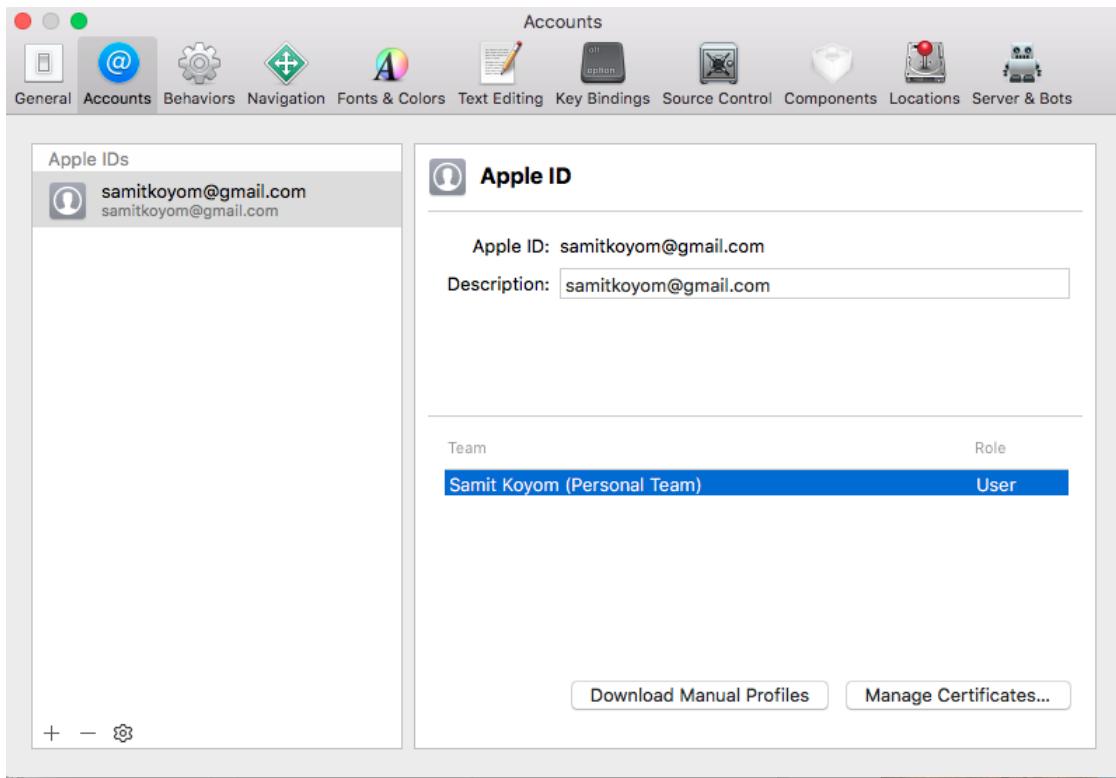
การติดตั้ง XCode และ Simulator



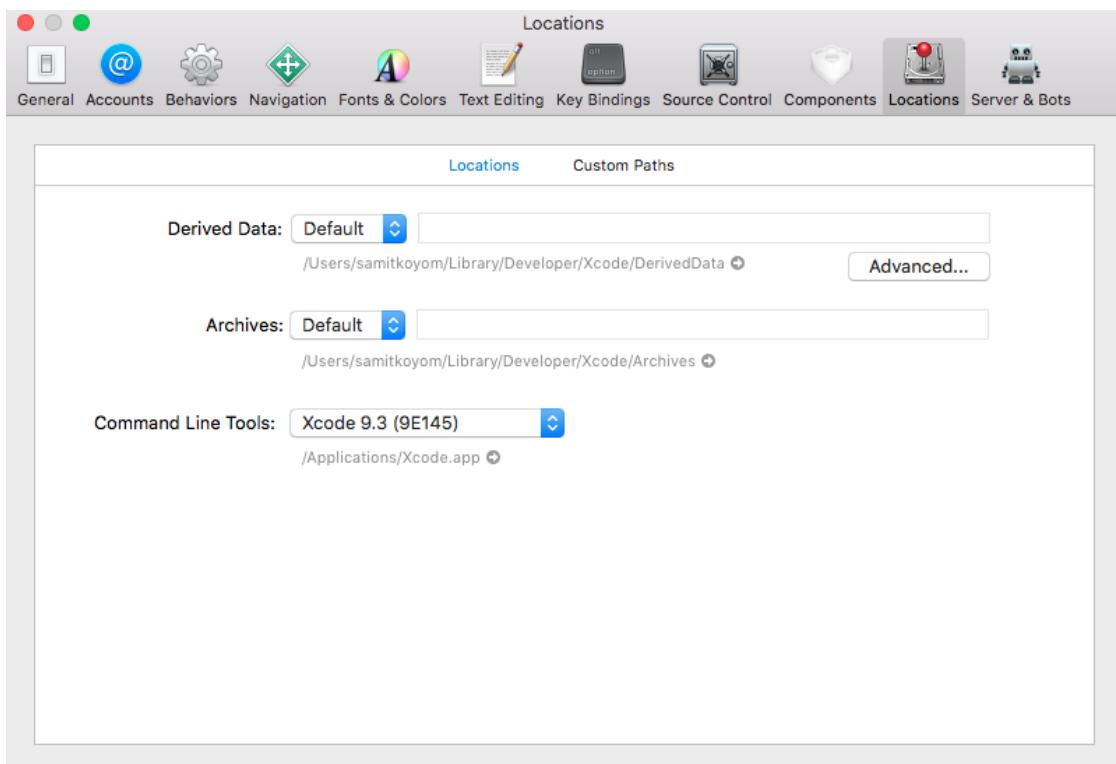
เข้า App Store ค้นหา Xcode และทำการกดที่ปุ่ม Get เพื่อเริ่มติดตั้ง ไฟล์มีขนาดประมาณ 5.2 GB อาจใช้เวลาค่อนข้างนาน



หลังจากติดตั้งแล้วเสร็จเปิด โปรแกรม Xcode ขึ้นมาแล้ว เข้าไปที่เมนู Xcode > Preferences... จากนั้นทำการเพิ่ม Apple ID ของคุณลงไปดังภาพ

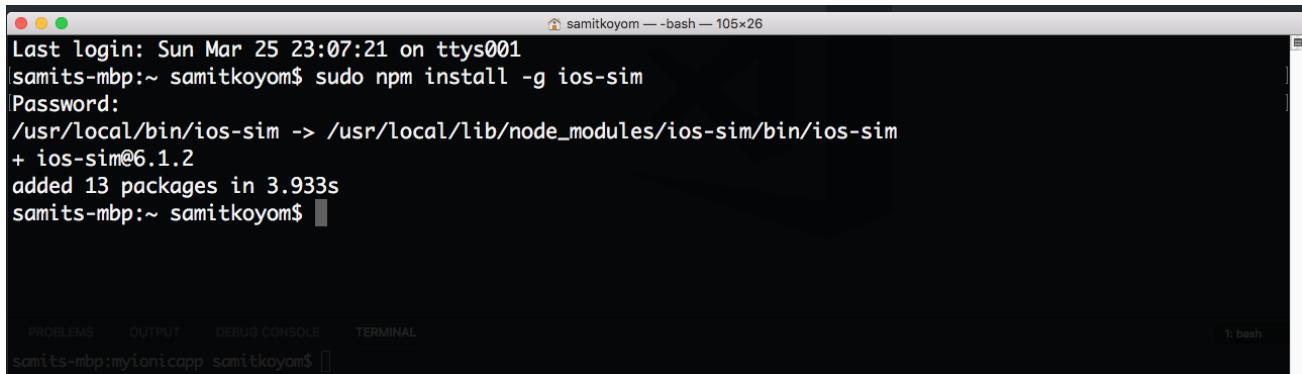


กดเพิ่ม Team ด้านล่างดังภาพด้วย



จากนั้นไปที่แท็บ Location เลือกตั้งค่าตรง Command Line Tools เป็นรุ่น Xcode ที่คุณติดตั้งดังภาพ เป็นอันเรียบร้อย

การติดตั้ง ios-sim สำหรับรัน ios ผ่าน command line

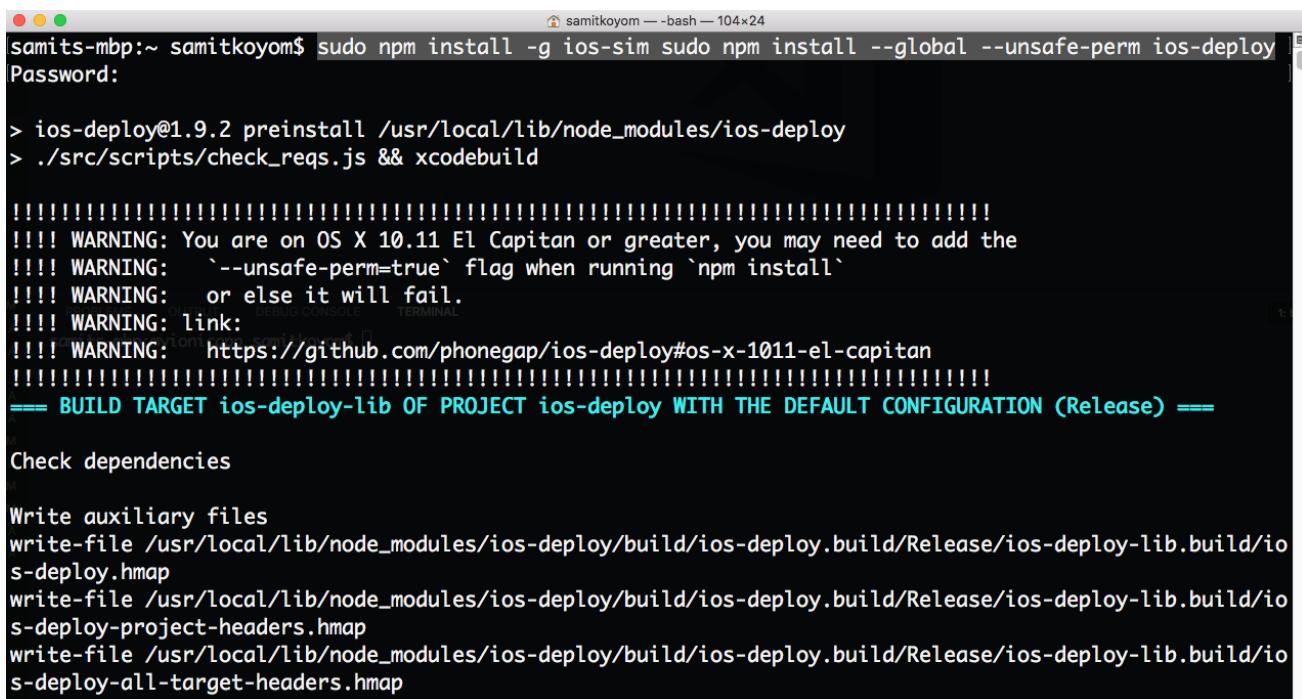


```
Last login: Sun Mar 25 23:07:21 on ttys001
samits-mbp:~ samitkoyom$ sudo npm install -g ios-sim
Password:
/usr/local/bin/ios-sim -> /usr/local/lib/node_modules/ios-sim/bin/ios-sim
+ ios-sim@6.1.2
added 13 packages in 3.933s
samits-mbp:~ samitkoyom$
```

The screenshot shows a macOS terminal window titled "samitkoyom — bash — 105x26". It displays the command `sudo npm install -g ios-sim` being run, along with its output. The terminal interface includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL, with the current tab being TERMINAL.

เปิด Terminal พิมพ์คำสั่ง sudo npm install -g ios-sim

การติดตั้ง ion-deploy เพื่อรันและคอมไพล์ ios app



```
samits-mbp:~ samitkoyom$ sudo npm install -g ios-sim sudo npm install --global --unsafe-perm ios-deploy
Password:

> ios-deploy@1.9.2 preinstall /usr/local/lib/node_modules/ios-deploy
> ./src/scripts/check_reqs.js && xcodebuild

!!!!!
!!!! WARNING: You are on OS X 10.11 El Capitan or greater, you may need to add the
!!!! WARNING: `--unsafe-perm=true` flag when running `npm install`
!!!! WARNING: or else it will fail.
!!!! WARNING: link:
!!!! WARNING: https://github.com/phonegap/ios-deploy#os-x-1011-el-capitan
!!!!!
== BUILD TARGET ios-deploy-lib OF PROJECT ios-deploy WITH THE DEFAULT CONFIGURATION (Release) ==

Check dependencies

Write auxiliary files
write-file /usr/local/lib/node_modules/ios-deploy/build/ios-deploy.build/Release/ios-deploy-lib.build/ios-deploy.hmap
write-file /usr/local/lib/node_modules/ios-deploy/build/ios-deploy.build/Release/ios-deploy-lib.build/ios-deploy-project-headers.hmap
write-file /usr/local/lib/node_modules/ios-deploy/build/ios-deploy.build/Release/ios-deploy-lib.build/ios-deploy-all-target-headers.hmap
```

The screenshot shows a macOS terminal window titled "samitkoyom — bash — 104x24". It displays the command `sudo npm install -g ios-sim sudo npm install --global --unsafe-perm ios-deploy` being run, along with its output. The terminal interface includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL, with the current tab being TERMINAL.

เปิด Terminal พิมพ์คำสั่ง

sudo npm install -g ios-sim sudo npm install --global --unsafe-perm ios-deploy

รอนานแล้วเสร็จเป็นอันเรียบร้อยสำหรับการเตรียมเครื่องบน mac เพื่อพัฒนา ios app

Module 3: พื้นฐาน JavaScript ES2015 (ES6) สำหรับ Angular และ Ionic

AGENDA

Classes

Modules

Consts

Arrow functions

Let

String Templates

Parameters

Promises

Transpilation



Destructuring

ความเปลี่ยนแปลงใหม่ ๆ ที่สำคัญใน JavaScript ES2015 หรือ ES6 ที่เราควรทำความเข้าใจเพื่อประกอบการใช้งาน Angular และ Ionic มีดังนี้

- ไม่จำเป็นต้องใส่ Semicolon (เซมิโคลอน) ท้ายคำสั่งก็สามารถทำงานได้แล้ว เช่น

```
function getMoneyFromGov() {
    var money = 500
    console.log("This "+moneny+" baht for new year gift")
}
```

- ใช้ let และ const ในการประกาศตัวแปรทดแทน var ได้

var มีปัญหาเดิมคือตัวแปรหลุ block การทำงานไปได้ ส่วน let และ const จะทำงานอยู่ภายใน scope มันเท่านั้น

```

let province = "Bangkok"
let district = "Donmuang"
let priceTag = 300.79

const PI = 3.14
const system_ver = 3.0

```

3. กำหนดค่าเริ่มต้น (default parameter) ให้ฟังก์ชันได้แล้ว
ปัญหาเดิมที่พบเจอ

```

function sayHi(name) {
    if(name==undefined) {
        name = "Somchai"
    }
    console.log("Hello "+name)
}

sayHi()

```

แก้ได้ด้วยการใช้ค่าเริ่มต้นให้ฟังก์ชันดังนี้

```

function sayHi(name="Chai"){
    console.log("Hello "+name)
}

sayHi()

```

4. Template String ช่วยให้เราทำการต่อข้อความกับตัวแปรได้ง่ายขึ้นมาก เช่น
แบบเดิม

```

var htmlContent = '<!DOCTYPE html>\n'+
    '<html>\n'+
    '<head>\n'+
    '<title>Home page</title>\n'+
    '</head>\n'+
    '<body>\n'+
    '<h1>Welcom to this website</h1>\n'+
    '</body>\n'+
    '</html>\n';

```

แบบใหม่

```
var htmlContent = `<!DOCTYPE html>
<html>
<head>
<title>Home page</title>
</head>
<body>
<h1>Welcom to this website</h1>
</body>
</html>`
```

สามารถแทรกรหัส HTML ได้ดังนี้

```
function printInfo(name, position, company) {
    console.log(` ${name} work as ${position} at ${company}`)
}
```

5. Arrow Function in ES6

เปลี่ยนจากเดิมเวลาประกาศฟังก์ชันจะใช้คีย์เวิร์ด function ใน ES2015 (ES6) จะทำการลดรูปเหลือเพียงเครื่องหมายลูกศรส่องเส้น หรือ fat arrow ดังนี้

แบบเดิม

```
var greet = function(name, message) {
    return message+name
}
greet(" Samit","Hello") // "Hello Samit"
```

แบบใหม่

```
var greet = (name,message) => {
    return message+name
}

greet(" Samit","Hello") // "Hello Samit"
```

หรือถ้าฟังก์ชันสั้น มีแค่การ return อย่างเดียวแบบตัวอย่างสามารถลดรูปได้แบบนี้

```
var greet = (name,message) => message+name
greet(" Samit","Hello") // "Hello Samit"
```

หรือหากฟังก์ชันรับพารามิเตอร์ มาแค่เพียงตัวเดียว ยังสามารถลดรูปต่อไปได้อีกดังนี้

```
var greet = message => message  
greet("Hello World") // "Hello World"
```

ตัวอย่างที่ Jeoboy ๆ เช่น

```
var square = x => x * x  
square(3) // 9
```

Module 4: พื้นฐาน Angular , Typescript และ Sass สำหรับ Ionic



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

การติดตั้ง Angular

A screenshot of a Mac OS X terminal window titled 'samitkoyom — bash — 76x15'. The window shows the command 'sudo npm install -g @angular/cli' being typed into the input field. The terminal interface includes standard Mac OS X window controls (red, yellow, green) and scroll bars.

เปิด Terminal หรือ Command Prompt ขึ้นมาพิมพ์คำสั่ง

บน Windows พิมพ์ npm install -g @angular/cli

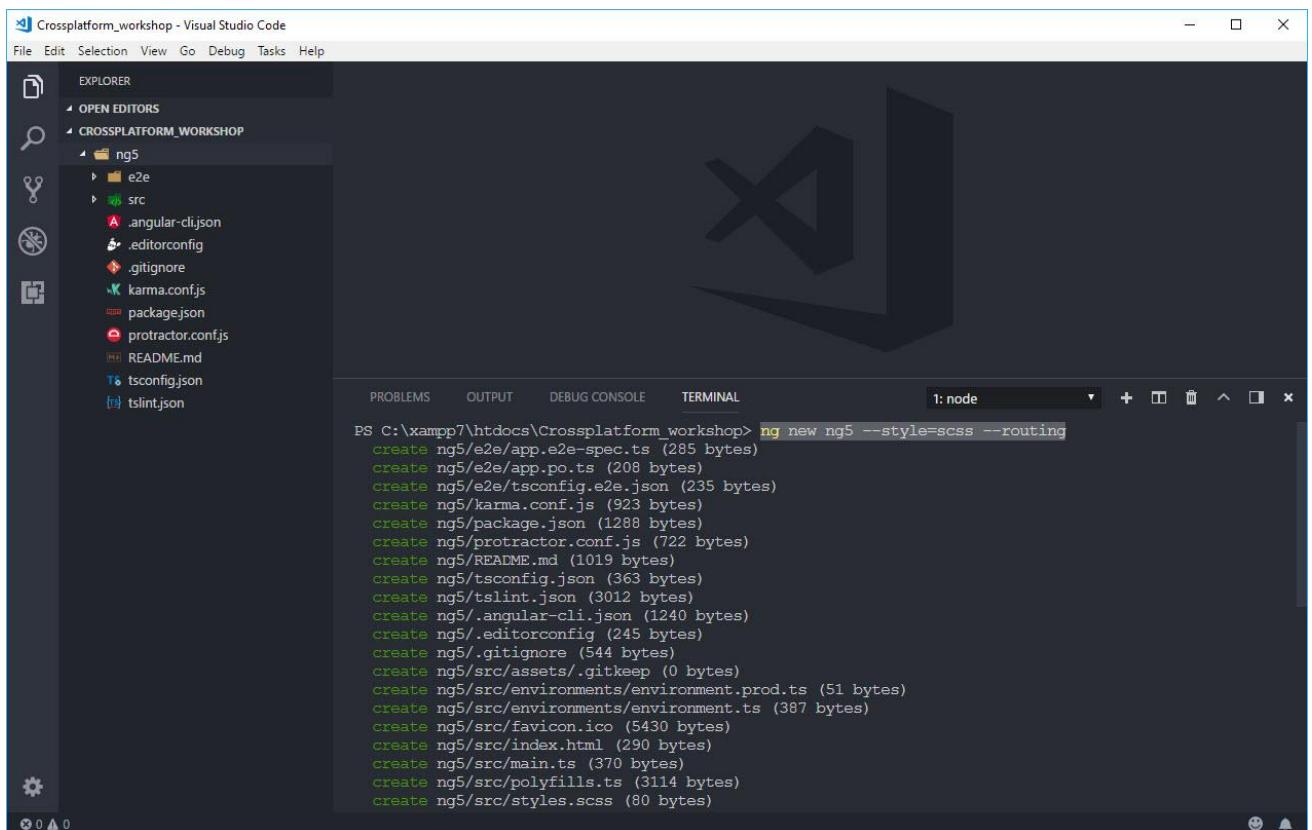
บน Mac พิมพ์ sudo npm install -g @angular/cli



```
samits-mbp:~ samitkoyom$ ng -v
Angular CLI: 1.7.3
Node: 8.10.0
OS: darwin x64
Angular:
...
samits-mbp:~ samitkoyom$
```

เมื่อติดตั้งเรียบร้อยแล้ว พิมพ์คำสั่ง `ng -v` เพื่อตรวจสอบเลข version จะแสดงข้อมูลคล้ายดังภาพ

สร้างโปรเจกต์แรกด้วย Angular กัน



Crossplatform_workshop - Visual Studio Code

File Edit Selection View Go Debug Tasks Help

EXPLORER

- OPEN EDITORS
- CROSSPLATFORM_WORKSHOP
 - ng5
 - e2e
 - src
 - .angular-cli.json
 - .editorconfig
 - .gitignore
 - karma.conf.js
 - package.json
 - protractor.conf.js
 - README.md
 - tsconfig.json
 - tslint.json

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\xampp7\htdocs\Crossplatform_workshop> `ng new ng5 --style=scss --routing`

```
create ng5/e2e/app.e2e-spec.ts (285 bytes)
create ng5/e2e/app.po.ts (208 bytes)
create ng5/e2e/tsconfig.e2e.json (235 bytes)
create ng5/karma.conf.js (923 bytes)
create ng5/package.json (1288 bytes)
create ng5/protractor.conf.js (722 bytes)
create ng5/README.md (1019 bytes)
create ng5/tsconfig.json (363 bytes)
create ng5/tslint.json (3012 bytes)
create ng5/.angular-cli.json (1240 bytes)
create ng5/.editorconfig (245 bytes)
create ng5/.gitignore (544 bytes)
create ng5/src/assets/.gitkeep (0 bytes)
create ng5/src/environments/environment.prod.ts (51 bytes)
create ng5/src/environments/environment.ts (387 bytes)
create ng5/src/favicon.ico (5430 bytes)
create ng5/src/index.html (290 bytes)
create ng5/src/main.ts (370 bytes)
create ng5/src/polyfills.ts (3114 bytes)
create ng5/src/styles.scss (80 bytes)
```

เปิดโปรเจกต์ Crossplatform_workshop ที่สร้างไว้ก่อนหน้านี้ขึ้นมาใน Visual Studio Code จากนั้นพิมพ์คำสั่งใน Terminal ของ VS Code ดังนี้

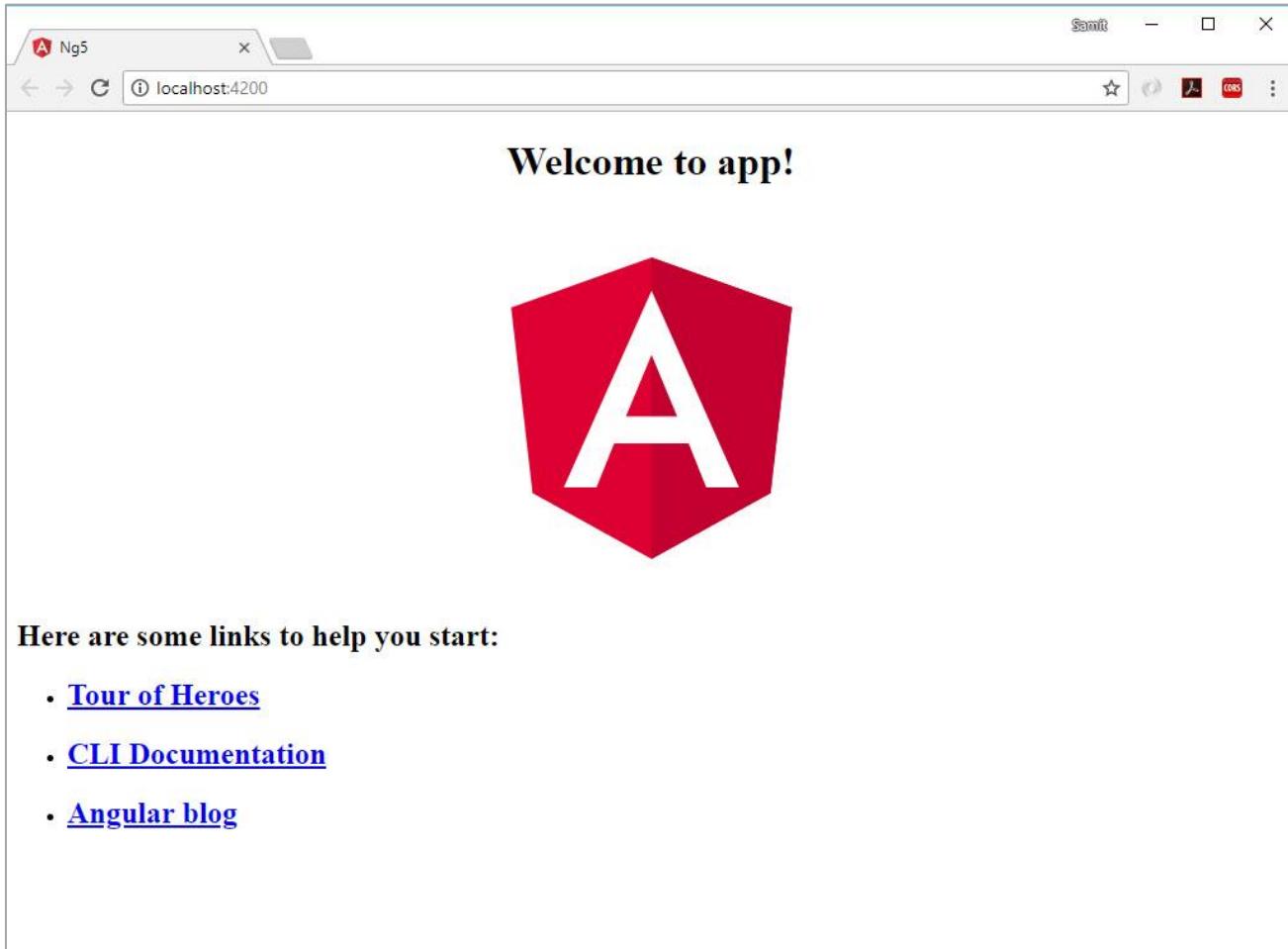
`ng new ng5 --style=scss --routing`

ร่องการดาวน์โหลดไฟล์ต่าง ๆ แล้วเสร็จ จากนั้นพิมพ์คำสั่งเปลี่ยน path เข้าไปในโปรเจกต์

```
cd ng5
```

จากนั้นทำการทดสอบบรันโปรเจกต์ไปแสดงบนหน้าเว็บ browser ด้วยคำสั่งด้านล่างนี้

```
ng serve --open
```



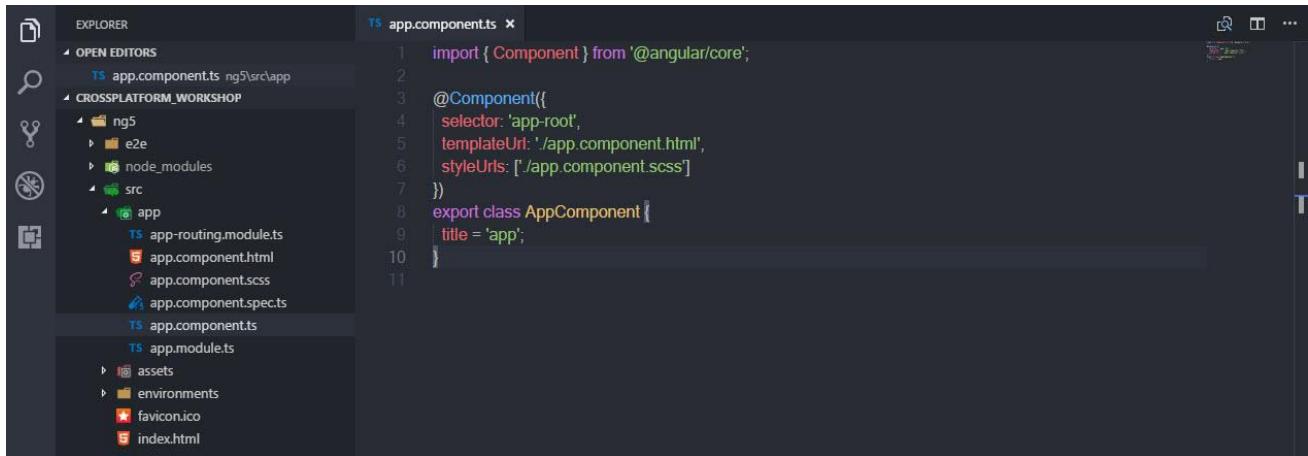
หากถูกต้องจะมีการแสดงหน้าจอ Welcome to app บน browser ที่ port 4200 ดังภาพ

Angular 5 Components

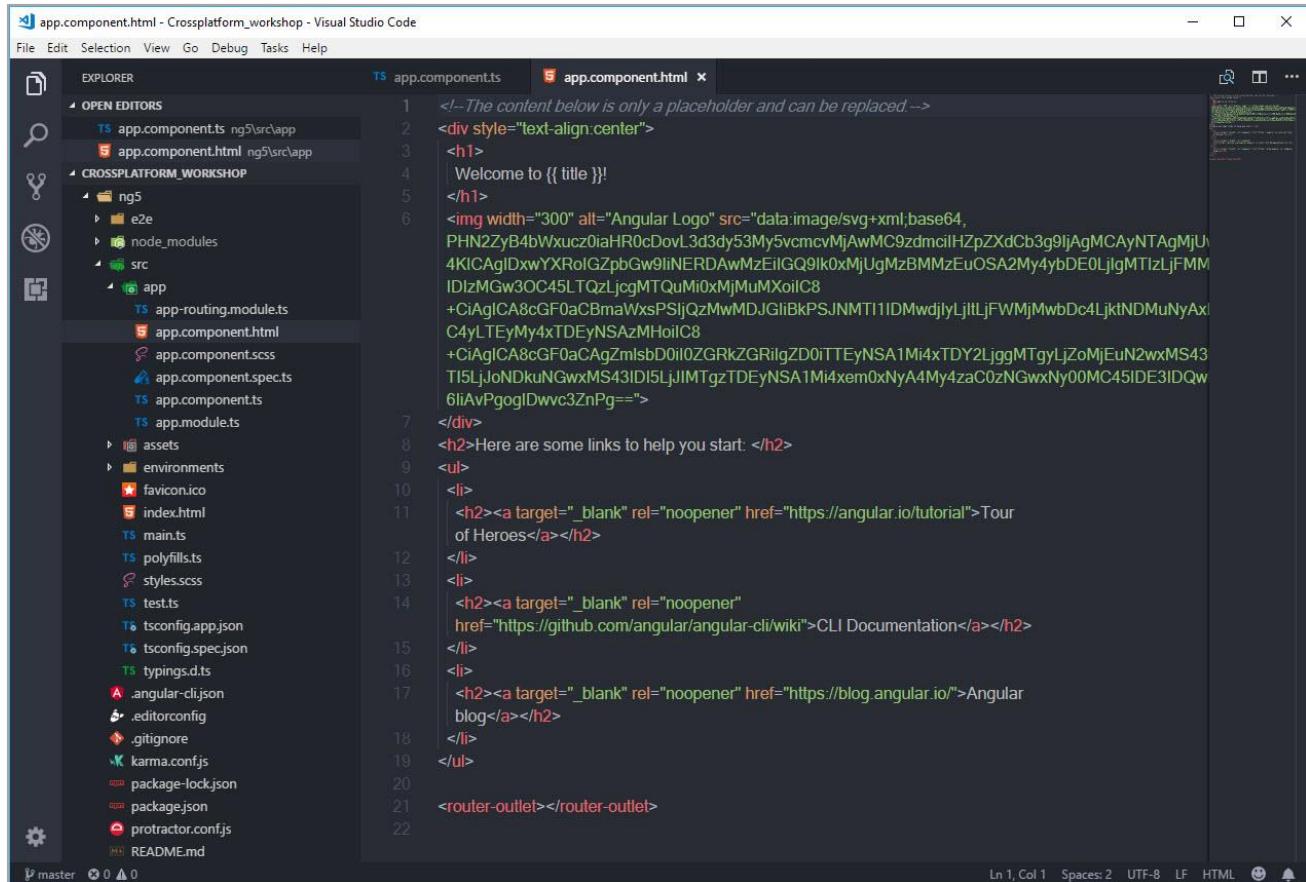
Angular components are the basic building blocks of your app. Each component defines:

- Any necessary imports needed by the component
- A component decorator, which includes properties that allow you to define the template, CSS styling, animations, etc..
- A class, which is where your component logic is stored.

Angular components reside within the `/src/app` folder:



```
import { Component } from '@angular/core';
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent {
  title = 'app';
}
```



```

<!--The content below is only a placeholder and can be replaced.-->
<div style="text-align:center">
<h1>
  Welcome to {{ title }}!
</h1>

</div>
<h2>Here are some links to help you start:</h2>
<ul>
<li>
<h2><a target="_blank" rel="noopener" href="https://angular.io/tutorial">Tour of Heroes</a></h2>
</li>
<li>
<h2><a target="_blank" rel="noopener" href="https://github.com/angular/angular-cli/wiki">CLI Documentation</a></h2>
</li>
<li>
<h2><a target="_blank" rel="noopener" href="https://blog.angular.io/">Angular blog</a></h2>
</li>
</ul>
<router-outlet></router-outlet>

```

Ln 1, Col 1 Spaces: 2 UTF-8 LF HTML ☺ 🔔

การสร้าง Component ใหม่

ng generate component home

หรือเขียนแบบบย่อ ng g c home

สร้าง component ชื่อ about

ng g c about

```
** NG Live Development Server is listening on localhost:4200, open your browser to http://localhost:4200
PS C:\xampp7\htdocs\Crossplatform workshop\ng5> ng g c home
  create src/app/home/home.component.html (23 bytes)
  create src/app/home/home.component.spec.ts (614 bytes)
  create src/app/home/home.component.ts (262 bytes)
  create src/app/home/home.component.scss (0 bytes)
  update src/app/app.module.ts (469 bytes)
PS C:\xampp7\htdocs\Crossplatform_workshop\ng5> ng g c about
  create src/app/about/about.component.html (24 bytes)
  create src/app/about/about.component.spec.ts (621 bytes)
  create src/app/about/about.component.ts (266 bytes)
  create src/app/about/about.component.scss (0 bytes)
  update src/app/app.module.ts (547 bytes)
PS C:\xampp7\htdocs\Crossplatform_workshop\ng5>
```

จากนั้นทำการเปิดไฟล์ /src/app/app.component.html

```
<ul>
  <li><a routerLink="home">Home</a></li>
  <li><a routerLink="about">About</a></li>
</ul>

<app-home></app-home>

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

เพิ่มคำสั่งลงไปดังภาพ

Angular 5 Template & Styling

ເປີດໄຟລ໌ [home.component.html](#) ເພີ່ມໂຄດດັ່ງນີ້

```
<div class="container color-dark">

  <div class="col">
    <p>Add a bucket list item</p>
  </div>

  <div class="col">
    <p>Your bucket list</p>
  </div>

</div>

<div class="container color-light">

  <div class="col">
    <p class="sm">Use this form below to add a new bucket list goal. What do you want to accomplish in your life?</p>

    <form>
      <input type="text" class="txt" name="item" placeholder="Life goal...">
      <input type="submit" class="btn" value="Add Item">
    </form>
  </div>

  <div class="col">
    <p class="life-container">
      I want to climb a mountain
    </p>
  </div>
</div>
```

ເປີດໄຟລ໌ /src/styles.scss ເພີ່ມໂຄິດດັ່ງນີ້

```
@import url('https://fonts.googleapis.com/css?family=Raleway:300,700');

body {
    background:#2E9CE6;
    padding: 3em;
    font-family: 'Raleway', 'Arial';
    color:#fff;
}

ul {
    list-style-type:none;
    margin:0 0 2em 0;
    padding:0;
}

ul li {
    display:inline;
    margin-right: 30px;
}

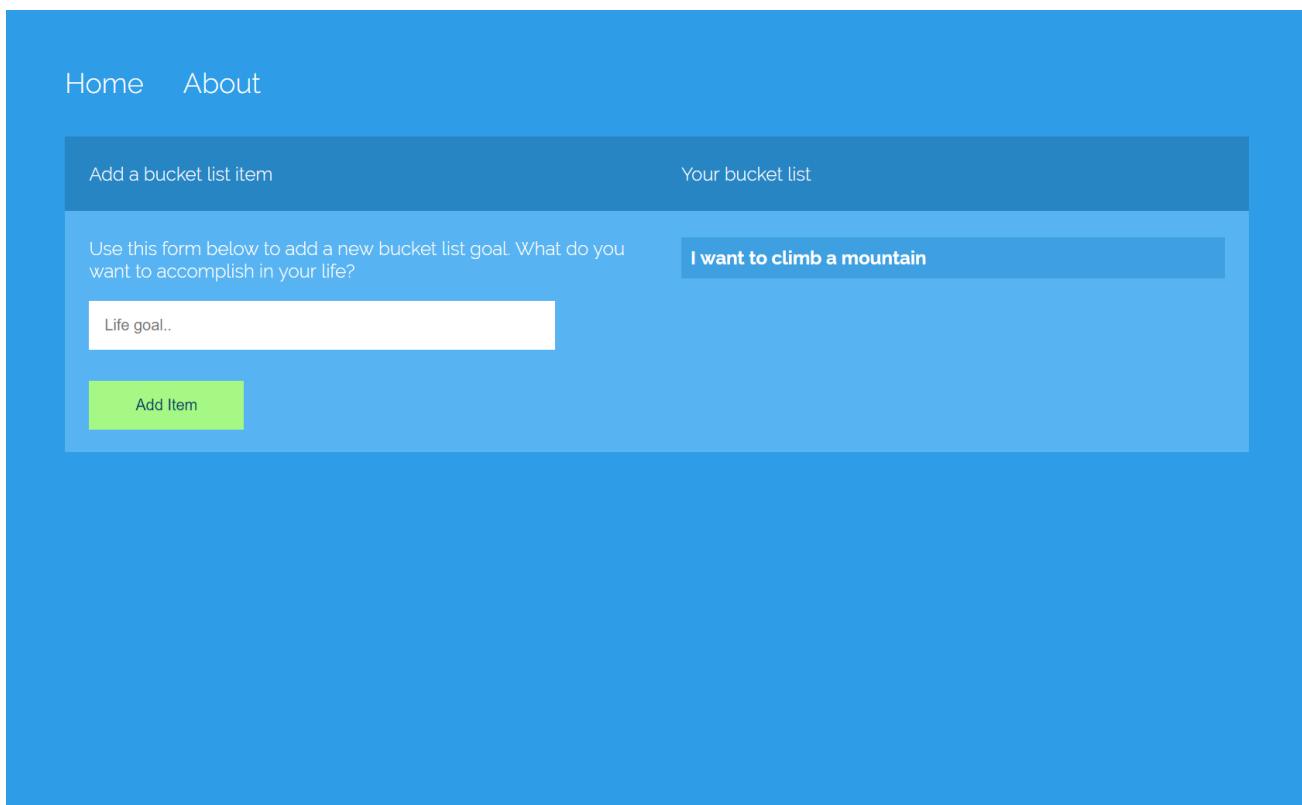
ul li a {
    font-size: 1.5em;
}

a {
    color:#fff;
    text-decoration:none;
}
```



```
.container {  
    display: grid;  
    grid-template-columns: 50% auto;  
}  
.col {  
    padding: .4em 1.3em;  
}  
.color-dark {  
    background: #2885C4;  
}  
.color-light {  
    background: #57B3F1;  
}  
input.txt {  
    border: 0;  
    padding: 1em;  
    width: 80%;  
    margin-bottom: 2em;  
}  
input.btn {  
    border: 0;  
    display: block;  
    padding: 1em 3em;  
    background: #A5F883;  
    color: #003A61;  
    margin-bottom: 1em;  
    cursor: pointer;  
}
```

```
.life-container {  
background:#3FA0E1;  
padding:.5em;  
font-weight:bold;  
cursor:pointer;  
}
```



ทดสอบแสดงผลจะได้ผลลัพธ์ดังภาพ

Angular 5 Interpolation, Property Binding & Event Binding

เปิดไฟล์ /src/app/home/home.component.ts เพิ่มตัวแปรดังนี้

```
export class HomeComponent implements OnInit {  
  itemCount: number = 4;  
  
  constructor() {}  
  
  ngOnInit() {}  
}
```

จากนั้นในไฟล์ /home.component.html วางตัวแปรลงไปดังนี้

```
<!-- From: -->  
  
<p>Your bucket list</p>  
  
<!-- To: -->  
  
<p>Your bucket list {{ itemCount }}</p>
```

Two-Way Data Binding

Open up /src/app/app.module.ts

```
import { FormsModule } from '@angular/forms';

// Other imports removed for brevity

@NgModule({
  ...
  imports: [
    BrowserModule,
    AppRoutingModule,
    FormsModule // Add the FormsModule here
  ],
})
```

กลับมาที่ไฟล์ /home.component.ts

```
itemCount: number = 4;
btnText: string = 'Add an Item';
goalText: string = 'My first life goal'; // Add this
```

ใน template file แก้ไขดังนี้

```
<!-- From: -->
<input type="text" class="txt" name="item" placeholder="Life goal..">
<!-- To: -->
<input type="text" class="txt" name="item" placeholder="Life goal.." [(ngModel)]="goalText">
<br><span>{{ goalText }}</span><br>
```

Event Binding

เพิ่มเหตุการณ์ (event) คลิกลงไปบนปุ่ม submit

```
<!-- From: -->  
  
<input type="submit" class="btn" value="{{ btnText }}>  
  
<!-- To: -->  
  
<input type="submit" class="btn" value="{{ btnText }}" (click)="addItem()>
```

เขียนฟังก์ชัน addItem ที่หน้า HomeComponent

```
export class HomeComponent implements OnInit {  
  
  itemCount: number = 4;  
  
  btnText: string = 'Add an Item';  
  
  goalText: string = 'My first life goal';  
  
  goals: [];  
  
  constructor() {}  
  
  ngOnInit() {  
  
    this.itemCount = this.goals.length;  
  
  }  
  
  addItem() {  
  
    this.goals.push(this.goalText);  
  
    this.goalText = "";  
  
    this.itemCount = this.goals.length;  
  
  }  
}
```

ทำการ Loop แสดงผล array ที่ View

```
<!-- From: -->

<p class="life-container">I want to climb a mountain</p>

<!-- To: -->

<p class="life-container" *ngFor="let goal of goals">
  {{ goal }}
</p>
```

Using the Angular 5 Router

ไปทำการเพิ่ม routing ที่ไฟล์ /src/app/app-routing.module.ts

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';
import { HomeComponent } from './home/home.component'; // Add this
import { AboutComponent } from './about/about.component'; //Add this

const routes: Routes = [
  {
    path: '',
    component: HomeComponent
  },
  {
    path: 'about',
    component: AboutComponent
  }
];
```

เข้าไปแก้ไฟล์ AboutComponent จากนั้นลบส่วนที่ไม่ต้องการออกใน app.component.html

```
<ul>
  <li><a routerLink="">Home</a></li>
  <li><a routerLink="about">About</a></li>
</ul>
<router-outlet></router-outlet>
```

Angular 5 Deployment Tutorial

คำสั่งในการ Build Project Angular

```
ng build
```

```
ng build --prod
```

```
ng build --prod --base-href="myURL"
```

Module 4: รู้จักกับ Ionic Framework

- แนะนำเวอร์ชันของ Ionic และความแตกต่าง
- ระบบและโปรแกรมที่ใช้ใน Ionic framework
- ทำความรู้จัก Ionic CLI
- สร้างโปรเจกต์แรกใน Ionic framework
- ทำความเข้าใจโครงสร้างของโปรเจค Ionic
- คำสั่งรัน Web server: ionic serve
- ทดสอบใช้งาน Ionic Serve
- วิธีแสดงแอพแบบ 2 แพลตฟอร์มบน Ionic Serve

การสร้างโปรเจกต์ใหม่ ใน ionic framework

1. ionic start ชื่อโปรเจกต์ ชื่อเทมเพลต

เทมเพลตมี 3 แบบ

- blank

- sidemenu

- tabs

ionic start tabsapp tabs

2. cd tabsapp

3. สร้างขึ้น browser

ionic serve

ionic serve -l

4. เพิ่ม platform android

ionic cordova platform add android

5. ทำการ build เป็น แอพ android

ionic cordova build android

6. ทำการรันขึ้นมือถือ

ionic cordova run android

Module 5: การสร้างหน้า Page , Basic Navigation และการส่งข้อมูลระหว่างเพจ

คำสั่งสร้างหน้าจอใหม่ใน Ionic framework

ionic g page schedule

ionic g page portfolio

ionic g page payment

ionic g page setting

การเปิดหน้า page ของ ionic สามารถทำได้ 2 วิธี

1. แบบเปิดแทนที่หน้าเดิม

this.nav.setRoot(หน้าที่ต้องการเปิด)

2. แบบเปิดขึ้นมาซ้อนหน้าเดิม

this.nav.push(หน้าที่ต้องการเปิด)

การสร้าง Template App แบบ Side Menu ร่วม กับ Tabs

ปกติการสร้างโปรเจกต์แบบ Ionic จะมี Template ให้เลือกอยู่ 3 แบบด้วยกัน นั่นคือ แบบ blank , sidemenu และ tabs โดยหากนักพัฒนาต้องการเมนูทั้งแบบ sidemenu และ Tabs รวมเข้าด้วยกัน สามารถทำได้ดังนี้

STEP 1: สร้างโปรเจกต์ใหม่

ในขั้นตอนการสร้างโปรเจกต์ใหม่ แนะนำให้เลือก template เป็นแบบ sidemenu ไว้ก่อนเลย และเราจะค่อยเพิ่มแบบ tabs เข้าไปทีหลัง

ionic start combindapp sidemenu

หลังจากติดตั้งโปรเจกต์เสร็จเรียบร้อยแล้ว เราจะทำการลบหน้า home , list ออกไปก่อน

STEP 2: สร้างหน้าสำหรับ Side Menu

ในขั้นตอนนี้เราจะมาสร้างหน้าสำหรับไว้เป็นเมนูด้านข้างกัน โดยจะทดสอบสร้างมา 4 เมนู ดังนี้

ionic g page side_schedule

ionic g page side_portfolio

```
ionic g page side_payment
ionic g page side_setting
```

STEP 3: สร้างหน้าสำหรับ Tabs Menu

ในขั้นตอนนี้จะมาสร้างหน้าสำหรับไว้แสดง Tabs เม뉴 โดยจะทดสอบสร้างมา 5 หน้าด้วยกันดังนี้

```
ionic g page tab_home
ionic g page tab_course
ionic g page tab_service
ionic g page tab_article
ionic g page tab_contact
```

STEP 4: สร้างหน้า page สำหรับรวม Tabs เข้าด้วยกัน

```
ionic g page tabs
```

STEP 5: เขียนคำสั่งในหน้า src/pages/tabs/tabs.ts ดังนี้

```
export class TabsPage {
  tab1Root: any = TabHomePage;
  tab2Root: any = TabCoursePage;
  tab3Root: any = TabServicePage;
  tab4Root: any = TabArticlePage;
  tab5Root: any = TabContactPage;
  ...
  ...
}
```

STEP 6: เขียนคำสั่งในหน้า src/pages/tabs/tabs.html ดังนี้

```
<ion-tabs>
  <ion-tab tabIcon="home" [root]="tab1Root" tabTitle="หน้าหลัก"></ion-tab>
  <ion-tab tabIcon="book" [root]="tab2Root" tabTitle="คอร์ส"></ion-tab>
  <ion-tab tabIcon="heart" [root]="tab3Root" tabTitle="บริการ"></ion-tab>
  <ion-tab tabIcon="chatboxes" [root]="tab4Root" tabTitle="บทความ"></ion-tab>
  <ion-tab tabIcon="mail-open" [root]="tab5Root" tabTitle="ติดต่อเรา"></ion-tab>
</ion-tabs>
```

STEP 7: เข้าหน้า src/app/app.modules.ts

ทำการลบส่วนของ

```
import { HomePage } from './pages/home/home';
import { ListPage } from './pages/list/list'; ออกไปก่อน
```

STEP 8: เข้าหน้า src/app/app.component.ts

ทำการลบส่วนของ

```
import { HomePage } from './pages/home/home';
import { ListPage } from './pages/list/list'; ออกไปก่อน
```

แก้ไขส่วนของ

```
rootPage: any = HomePage;
```

เป็น

```
rootPage: any = TabsPage;
```

แก้ไขส่วน

```
this.pages = [
  { title: 'Home', component: HomePage },
  { title: 'List', component: ListPage }
];
```

เป็น

```
this.pages = [
  { title: 'ตารางอบรม', component: SideSchedulePage },
  { title: 'ผลงานของเรา', component: SidePortfolioPage },
  { title: 'ช่องทางชำระเงิน', component: SidePaymentPage },
  { title: 'ตั้งค่าระบบ', component: SideSettingPage }
];
```

แก้ไขส่วนของ

```
openPage(page) {
  // Reset the content nav to have just this page
```

```

// we wouldn't want the back button to show in this scenario
this.nav.setRoot(page.component);

}

เป็น

openPage(page) {
  // Reset the content nav to have just this page
  // we wouldn't want the back button to show in this scenario
  this.nav.push(page.component);
}

```

STEP 9: เพิ่มส่วนของ Hamberger menu ให้กับหน้าที่เป็น tabs ทั้ง 5 หน้าดังนี้
เข้าไปที่ src /pages/tab-menu1/tab-menu1.html และใส่คำสั่งดังนี้

```

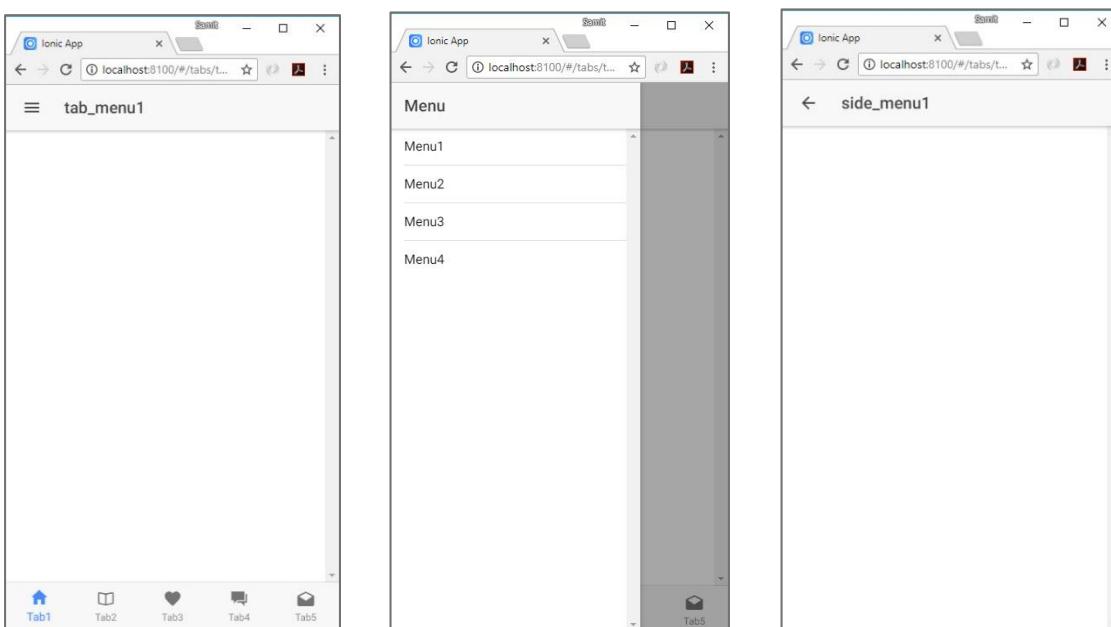
<ion-navbar>
  <button ion-button menuToggle="left" left>
    <ion-icon name="menu"></ion-icon>
  </button>
  <ion-title>tab_menu1</ion-title>
</ion-navbar>

```

STEP 10: ทดสอบรันโปรแกรม

ionic serve

น่าจะได้ผลลัพธ์ดังภาพ



การสร้างเพิ่ม Side Menu ด้านขวาและการ Custom Nabbar menu

โปรเจกต์ ionic แบบ Sidemenu จะมีเมนูอยู่ด้านซ้าย เพียงชุดเดียว หากนักพัฒนาต้องการเมนูเพิ่มเติม ด้านขวา หรือหากต้องการย้ายชุดเมนูไปวางไว้ทางขวา ก็สามารถทำได้ตามขั้นตอนในตัวอย่างนี้

STEP 1: ลองย้ายเมนูไปไว้ทางขวา

ในขั้นตอนนี้เรามาลองทำการเพิ่มคำสั่งเพื่อทำการย้ายเมนูไปไว้ทางด้านขวา กัน

เปิดไฟล์ src/app/app.html และทำการเพิ่มคุณสมบัติ side="right" ลงไปดังนี้

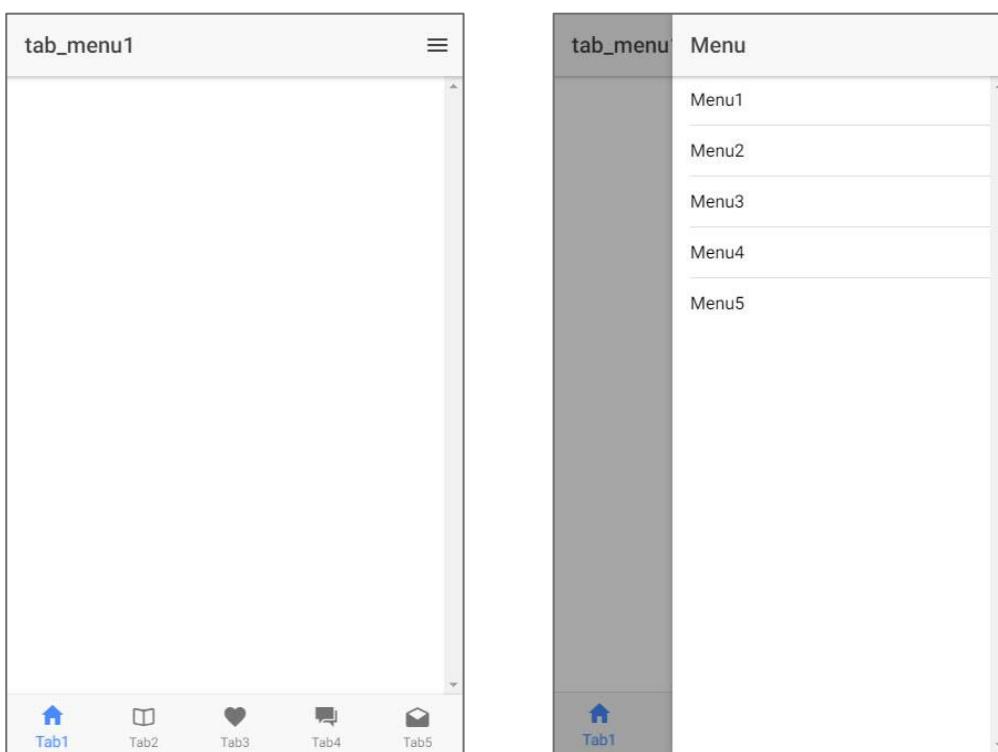
```
<ion-menu [content]="content" side="right">
```

STEP 2: เปิดหน้า tabs ต่าง ๆ เพื่อแก้ไขตำแหน่งดังนี้

เปิดไฟล์ src/pages/tab-menu1/tab-menu1.html และแก้คุณสมบัติดังนี้

```
<button ion-button menuToggle="right" right>
  <ion-icon name="menu"></ion-icon>
</button>
```

STEP 3: ทดสอบรันโปรเจกต์จะได้ผลลัพธ์ดังภาพ



STEP 4: ลองทำการเพิ่ม Tabs เมนูทั้ง 2 ข้าง (ซ้าย-ขวา)

เริ่มต้นจะขอป้ายเมนูชุดเดิมไว้ทางซ้ายเช่นเดิมก่อน

จากนั้นเปิดไฟล์ src/app/app.html และเพิ่มชุดคำสั่งเพื่อแสดงเมนูด้านขวาดังนี้

<!-- Right Menu-->

```
<ion-menu [content]="content" side="right" id="menu_right">
  <ion-content>
    <ion-list>
      <button menuClose ion-item>
        <ion-icon name="medical"></ion-icon> Setting 1
      </button>
      <button menuClose ion-item>
        <ion-icon name="medical"></ion-icon> Setting 2
      </button>
    </ion-list>
  </ion-content>
</ion-menu>
```

STEP 5: ทดสอบรันก็จะได้ผลลัพธ์ดังนี้



STEP 5: ปรับแต่ง Navbar (toolbar) ในแต่ละหน้าของ tabs เพื่อแสดง hamburger menu และเมนูอื่น ๆ เพิ่มเติม

เข้าไปที่หน้า src/pages/tab-menu1.html ทำการเพิ่มคำสั่งดังนี้

```
<ion-navbar>

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

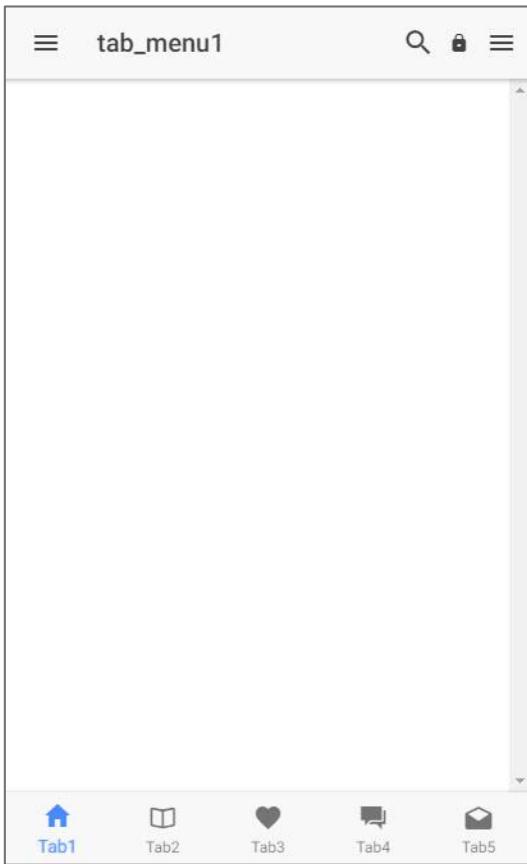
    <ion-title>tab_menu1</ion-title>

    <ion-buttons right>
        <button ion-button icon-only>
            <ion-icon name="search"></ion-icon>
        </button>
        <button ion-button>
            <ion-icon name="lock"></ion-icon>
        </button>
    </ion-buttons>

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

</ion-navbar>
```

STEP 6: ทดสอบรันแสดงผลลัพธ์ที่จะได้ดังภาพ



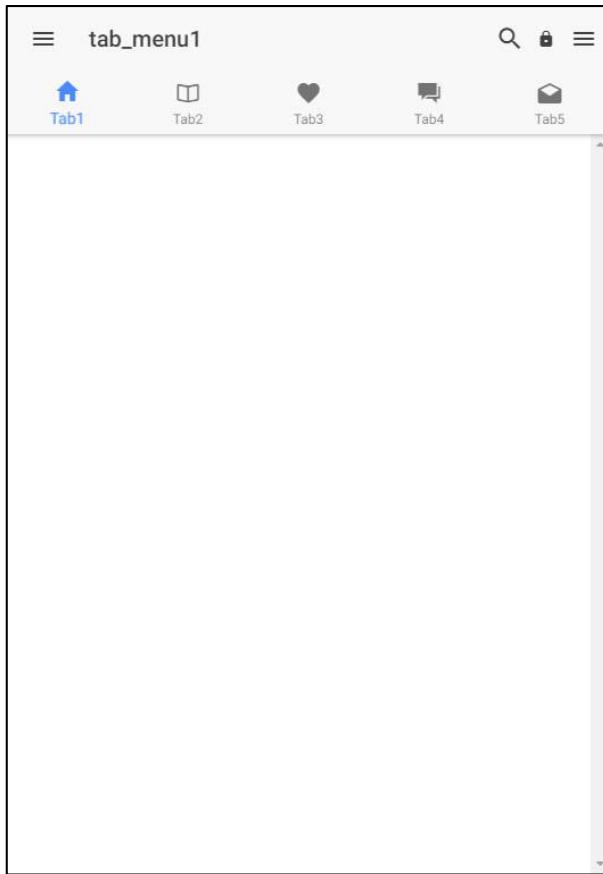
การย้าย Tabs menu ไว้ด้านบนและการทำ Slide tabs

ในตอนนี้เราจะมาทำการย้าย Tabs เมนูไปไว้ด้านบนกัน ซึ่งปกติโปรเจ็คต์ของ Ionic จะวาง tabs menu ไว้ด้านล่างตอนเริ่มต้น เพื่อให้ UX ดีขึ้นเราสามารถทำได้ดังนี้

STEP 1: การย้าย Tabs menu ไว้ด้านบน

```
เข้าไปแก้ไขไฟล์ src/app/app.module.ts ดังนี้
imports: [
  BrowserModule,
  IonicModule.forRoot(MyApp,{tabsPlacement: 'top'}),
  HttpModule
],
```

STEP 2: ทดสอบรัน ionic serve ดู น่าจะได้ผลลัพธ์ดังภาพ



สังเกตว่า Tabs เมนูอยู่ด้านบนเรียบร้อยแล้ว

การทำ Slide tabs เมนู

หากเราต้องการให้สามารถ Side tab เมื่อทำการปัดซ้ายขวาเราสามารถทำได้ดังนี้

STEP 1: ติดตั้งส่วนเสริม Native Page Transitions

ionic cordova plugin add com.telerik.plugins.nativepagetransitions

npm install --save @ionic-native/native-page-transitions

STEP 2: เข้าไปทำการ import ส่วนเสริม ที่ไฟล์ src/app/app.module.ts ดังนี้

```
import { NativePageTransitions } from '@ionic-native/native-page-transitions';
```

```
providers: [
```

```
StatusBar,
```

```
SplashScreen,
```

```
{provide: ErrorHandler, useClass: IonicErrorHandler},
NativePageTransitions
]
```

STEP 3: เข้าไปทำการเพิ่มฟังก์ชันในไฟล์ src/pages/tabs/tabs.ts

```
import { NativePageTransitions, NativeTransitionOptions } from '@ionic-native/native-
page-transitions';

export class TabsPage {

loaded: boolean = false;
tabIndex: number = 0;

.

.

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

}
```

สร้างฟังก์ชันสำหรับการ swipe 2 ฟังก์ชันคือ getAnimationDirection() และ transition()

```
private getAnimationDirection(index):string {
  var currentIndex = this.tabIndex;
```

```
this.tabIndex = index;

switch (true){
  case (currentIndex < index):
    return('left');
  case (currentIndex > index):
    return ('right');
}
```

```
}
```

```
public transition(e):void {
  let options: NativeTransitionOptions = {
    direction:this.getAnimationDirection(e.index),
```

```

duration: 0,
slowdownfactor: -1,
slidePixels: 0,
iosdelay: 20,
androiddelay: 0,
fixedPixelsTop: 0,
fixedPixelsBottom: 48
};

if (!this.loaded) {
  this.loaded = true;
  return;
}

this.nativePageTransitions.slide(options);
}

```

STEP 4: เข้าไปเพิ่ม event ในไฟล์ src/pages/tabs/tabs.html

```

<ion-tabs (ionChange)="transition($event)">
<ion-tab tabIcon="home" [root]="tab1Root" tabTitle="Home"></ion-tab>
.
.
</ion-tabs>

```

STEP 5: เข้าไปสร้างฟังก์ชันการ swipe ที่ไฟล์ tabs แต่ละหน้า เช่น src/pages/tab-menu1/ tab-menu1.ts

```

swipe(event) {
  // direction 2 คือเลื่อนขวา
  if(event.direction === 2) {
    this.navCtrl.parent.select(1);
  }
}

```

```
หรือหากมีทั้ง swipe 左 และขวา
// direction 2 คือเลื่อนขวา
if(event.direction === 2) {
    this.navCtrl.parent.select(4);
}
```

```
// direction 4 คือเลื่อนซ้าย
if(event.direction === 4) {
    this.navCtrl.parent.select(2);
}
```

STEP 6: เข้าไปใส่ event ในหน้า tabs ที่เป็นส่วน html ทุกหน้า เช่น src/pages/tab-menu1/page-menu1.html

```
<ion-content>
    <div class="main-content" (swipe)="swipe($event)">
    </div>
</ion-content>
```

STEP 7: เข้าไปเพิ่ม style ที่ไฟล์ src/app/app.scss

```
.main-content {
    height: 100%;
}
```

STEP 8: ทดสอบ ionic serve และดูผลลัพธ์

Module 6: Ionic Components

- การออกแบบหน้าแอพพลิเคชันด้วย Component ที่ Ionic เตรียมไว้
- เรียนรู้การใช้งาน Component ขั้นประยุกต์
- การสร้างเพจแบบ Master Detail
- การใช้งาน HTTP Services และ Provider ใน Ionic
- การสร้างแบบฟอร์ม
- การตรวจสอบความถูกต้องของข้อมูล (Validations)
- การเพิ่มข้อมูลไปที่ส่วน Backend

การใช้งาน HTTP Services และ Provider ใน Ionic

พิมพ์คำสั่งสร้าง Provider ดังนี้

ionic generate provider webapi-service

เปิดไฟล์ src/app/app.module.ts เพิ่มคำสั่งดังนี้

```
// Web API
import { HttpModule } from '@angular/http';
import { WebapiServiceProvider } from './providers/webapi-service/webapi-service';
```

```
imports: [
  HttpModule
],
```

```
providers: [
  WebapiServiceProvider
]
```

ເປີດໄຟ້ src/providers/webapi-service/webapi-service.ts ຂຶ້ນມາເພີ່ມຄຳສັ່ງດັ່ງນີ້

```
import { Http, Headers } from '@angular/http';
import { Injectable } from '@angular/core';

@Injectable()
export class WebapiServiceProvider {

  // Base URL API
  baseUrl:any;

  constructor(public http: Http) {
    this.baseUrl = "http://192.168.1.51/url /";

    // POST Method
    postData(objdata, segment){
      return new Promise((resolve,reject)=>{
        let headers = new Headers();
        headers.append('Content-Type', 'application/json;charset=UTF-8');
        headers.append('Authorization', 'Basic c2FtaXQ6c21rMzc3MDQw');
        this.http.post(this.baseUrl+segment, JSON.stringify(objdata), {headers:headers})
        .subscribe(res => {
          resolve(res.json());
        }, (err)=> {
          reject(err);
        });
      });
    }
}
```

```
//GET METHOD

getData(segment){

    return new Promise((resolve,reject)=>{

        let headers = new Headers();

        headers.append('Content-Type', 'application/json;charset=UTF-8');

        headers.append('Authorization', 'Basic c2FtaXQ6c21rMzc3MDQw');

        this.http.get(this.baseUrl+segment, {headers:headers})

        .subscribe(res => {

            resolve(res.json());

        }, (err)=> {

            reject(err);

        });

    });

}

}
```

ตัวอย่างการเรียกใช้งาน

```
import { WebapiServiceProvider } from '../../providers/webapi-service/webapi-service';

this.webService.getData("users/list").then((result)=> {

    console.log(result);

    this.responseData = result;

}, (error)=> {

    this.responseNotFound = "มีบางอย่างผิดพลาดในการดึงข้อมูลจาก Server";

    console.log(error);

});
```

Module 7: การใช้งาน Git Version Control กับ Ionic

Step: 1

เช็คค่าลงโปรแกรม git ไว้หรือยัง

```
git --version
```

Step: 2

กำหนดข้อมูลประวัติผู้ใช้ (First-Time Git Setup)

```
$ git config --global user.name "John Doe"
```

```
$ git config --global user.email johndoe@example.com
```

Step: 3

ติดตั้ง git ลงในโปรเจกต์

```
git init
```

Step: 4

คำสั่งตรวจสอบสถานะไฟล์

```
git status
```

(สถานะไฟล์มี 3 แบบ

1. แบบยังไม่เก็บประวัติ (สีแดง)
2. เก็บประวัติแล้ว (สีเขียว)
3. บันทึกเข้าสู่ git แล้ว (จะหายไปจากคำสั่ง git status)

Step: 5 คำสั่งเก็บประวัติไฟล์

```
git add -A (ทั้งหมด)
```

```
git add .
```

Step: 6 บันทึกประวัติของไฟล์

```
git commit -m 'คำอธิบาย'
```

Step: 7 ทำการอัปโหลดไฟล์ขึ้น github (online)

```
git remote add origin https://github.com/iamsamitdev/ioniccombind.git
```

Step: 8 อัพโหลดไฟล์

```
git push -u origin master
```

```
git push --all
```

ย้อนประวัติของ git

```
git reset --hard commit_id (6 หลักแรก)
```

การดึงจาก remote ลงมาในเครื่องล่าสุด

```
git pull --all
```

การนำกลับมาใช้งาน

```
git clone https://github.com/iamsamitdev/ioniccombind.git
```

```
npm install
```

Module 8: การเก็บข้อมูลแบบ Local ด้วย Ionic Storage

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

constructor(
  public storage: Storage,) {
  this.initializeApp();

  }

this.storage.get('introShow').then((result) => {
  if (result) {
    // เคยเข้าใช้งานแล้ว
    this.rootPage = TabsPage;
    // เข้าใช้งานครั้งแรก
  } else {
    this.rootPage = IntroPage;
  }
});
```

Module 9: Ionic Native API

การเช็ค Network Detection with Ionic Native

แอพพลิเคชันส่วนใหญ่มีการเชื่อมต่อกับ web service และอาจต้องมีการต่อเน็ตไว้ก่อนเพื่อทำการเรียกใช้งาน บ่อยครั้งเมื่อผู้ใช้ offline เราต้องทำการตรวจสอบเสียก่อนจะทำงานส่วนต่อไปดังนี้ (ปล. พังก์ชันทำงานเฉพาะบน device จริงเท่านั้น)

STEP 1: ติดตั้งส่วนเสริม network connection ดังนี้

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

STEP 2: เข้าไปเพิ่มในไฟล์ src/app/app.module.ts ดังนี้

```
import { Network } from '@ionic-native/network';
@NgModule({
  //...
  providers: [
    Network
  ]
})
export class AppModule {}
```

STEP 3: เขียนคำสั่งตรวจเช็ค network ในหน้าที่ต้องการ เช่น src/page/tab-menu1/tab-menu1.ts ดังนี้

```
import { IonicPage, NavController, NavParams, ToastController } from 'ionic-angular';
import { Network } from '@ionic-native/network';

constructor(
  private toast: ToastController,
  private network: Network) {

}

// เช็คว่ามีการ disconnect ไปหรือเปล่า
if(this.network.type !== 'none'){

  ...
}
```

```
// เช็คว่ามีการ disconnect ไปหรือเปล่า
if(this.network.type !== 'none'){

  ...
}
```

```

}else if(this.network.type === 'none'){
  this.toast.create({
    message: 'Please Check your network and try again',
    duration: 3000
  }).present();

}else{
  this.toast.create({
    message: 'Please Check your network and try again',
    duration: 3000
  }).present();
}

```

การเช็ค Device เครื่อง

STEP 1: ติดตั้งส่วนเสริม

```

$ ionic cordova plugin add cordova-plugin-device
$ npm install --save @ionic-native/device

```

STEP 2: เพิ่มรายการส่วนเสริมเข้าหน้า `src/app/app.component.ts`

```

import { Device } from '@ionic-native/device';
providers: [
  ...
  Device,
  ...
]

```

STEP 3 : การเรียกใช้งาน

```
import { Device } from '@ionic-native/device';
constructor(private device: Device) { }

...
console.log('Device UUID is: ' + this.device.uuid);
```

Module 10: การทำ Push Notifications ใน Ionic

แอปส่วนใหญ่มีการทำ Push notification ไว้สำหรับส่งข่าวและแจ้งเตือนต่าง ๆ ให้ user ได้รับทราบ มีขั้นตอนการทำดังนี้

STEP 1: เข้าไปสร้างโปรเจกต์ใน <https://firebase.google.com/>

STEP 2 : ทำการดาวน์โหลดไฟล์ google-services.json และ GoogleService-Info.plist มาวางไว้ใน Root โฟลเดอร์ของโปรเจกต์

STEP 3: ติดตั้งส่วนเสริม fcm

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

STEP 4: เข้าไปทำการ Import ในไฟล์ src/app/app.module.ts

```
// Push Notification
import { FCM } from '@ionic-native/fcm';
import { Device } from '@ionic-native/device';

providers: [
  ...
  FCM,
  Device,
  ...
]
```

STEP 5: เข้าไปทำการเรียกใช้งาน Pushnotification ที่ไฟล์ src/app/app.component.ts

```
/** Push Notification */
import { FCM } from '@ionic-native/fcm';

constructor(
  private fcm: FCM,
) {
  this.initializeApp();
}

initializeApp() {
  this.platform.ready().then(() => {

    this.statusBar.styleDefault();
    this.splashScreen.hide();

    // เช็คว่าเป็น device จริงเท่านั้น
    if(!this.platform.is('core'))
    {
      // รับข้อมูลการ Pushnotification
      this.fcm.onNotification().subscribe(data=>{
        if(data.wasTapped){
          // alert('Recieve in background');
          this.nav.push('SideSchedulePage',{sid:data.pid});
        }else{

          //alert('Recieve in foreground');
          //alert(JSON.stringify(data));
          let alertdig = this.alertCtrl.create({
            title:data.title,
            subTitle:data.body,
            message:'pid='+data.pid+'groub='+data.groub,
            buttons:[

            ]
          }
        )
      })
    }
  })
}
```

```
        text: 'ចូរាយលេខេយ៍',
        handler: () => {
          this.nav.push('SideSchedulePage',{sid:data.pid});
        }
      },
    ],
    enableBackdropDismiss:false
  });

  alertdig.present();
}

});

// ចូរាយទៅក្នុង Token
this.fcm.onTokenRefresh().subscribe(token =>{
  //alert(token);
});

this.badge.set(10);

}
});

}


```

STEP 6: ដែល platform android (plugin fcm មិនបានអាក់បាន android ទៅពីរ ទៅពីរ នៅពេលដែលដោះស្រាយក្នុង android 6.4.0 ។
កំណត់)

ionic cordova platform add [android@6.4.0](#)

การเคลียร์ Notification ออกจาก tray ของ Android

STEP 1: เข้าไปเพิ่มคำสั่งในไฟล์ MainActivity.java (อยู่ที่ platforms/android/src/package_name/MainActivity.java)

```
package com.itgenius.combappdemo;
import android.os.Bundle;
import org.apache.cordova.*;
import android.app.NotificationManager;
import android.content.Context;

public class MainActivity extends CordovaActivity
{
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);

        // enable Cordova apps to be started in the background
        Bundle extras = getIntent().getExtras();
        if (extras != null && extras.getBoolean("cdvStartInBackground", false)) {
            moveTaskToBack(true);
        }

        final NotificationManager notificationManager = (NotificationManager)
this.getSystemService(Context.NOTIFICATION_SERVICE);
        notificationManager.cancelAll();

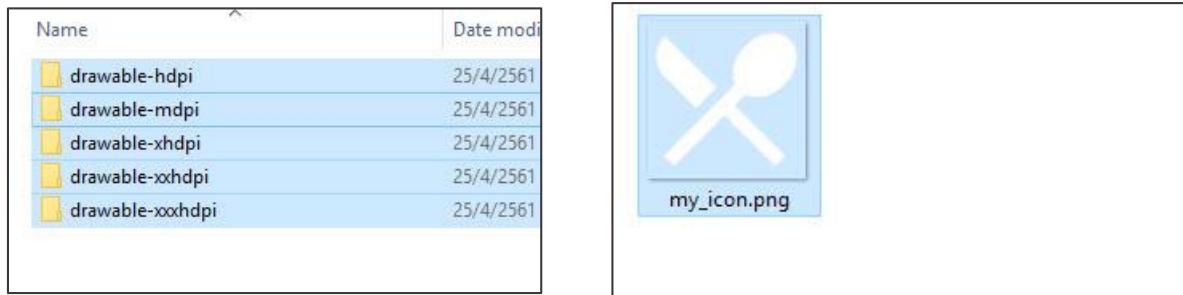
        // Set by <content src="index.html" /> in config.xml
        loadUrl(launchUrl);
    }

    @Override
    protected void onResume() {
        super.onResume();
        final NotificationManager notificationManager = (NotificationManager)
this.getSystemService(Context.NOTIFICATION_SERVICE);
        notificationManager.cancelAll();
    }
}
```

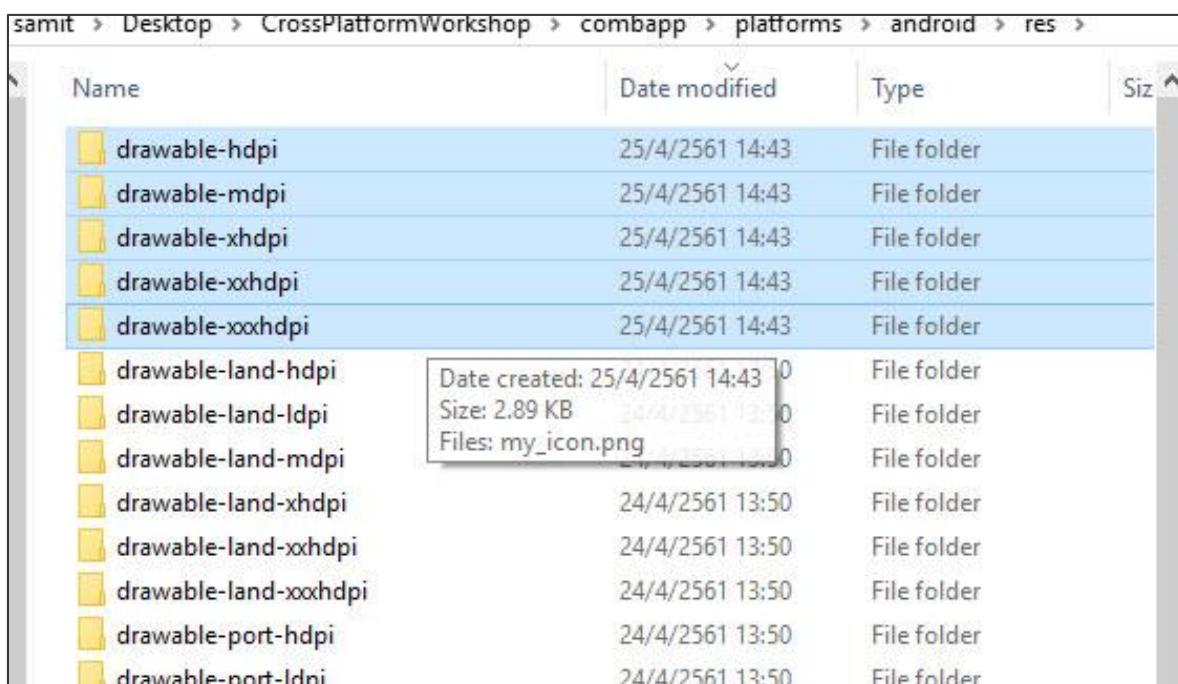
การเปลี่ยนรูป Icon ส่วน tray ของ notification

STEP 1: เตรียมรูปไอคอนเป็นแบบสีขาว (.png) ใช้บริการจากเว็บ

https://romannurik.github.io/AndroidAssetStudio/icons-notification.html#source.type=clipart&source.clipart=local_dining&source.space.trim=1&source.space.pad=0&name=ic_stat_local_dining



STEP 2: นำไฟล์รูปที่เตรียมมาทั้งหมดวางไว้ใน platforms/android/res



STEP 3: ตอนส่ง payload ให้ใส่ parameter นี้ไปด้วย “icon”:”my_icon”

```
10 //prep the bundle
11 $msg = array
12 (
13     'title'=> 'แจ้งข่าวอัพเดทระบบ',
14     'body'=> 'มีการอัพเดทระบบล่าสุดครั้งที่ ' . $pid .' ดูรายละเอียด',
15     'vibrate'=>true,
16     'sound'=>true,
17     'icon'=>'my_icon',
18     'badge'=> 5,
19     "click_action"=>"FCM_PLUGIN_ACTIVITY"
20 );
21
```

เรียบร้อยทดสอบ push และดูผลได้

Module 11: การทำแอพหลายภาษา (Multiple Language)

แบบ Lazy Load

STEP 1: ติดตั้ง ngx-translate ในโปรเจ็กต์

```
npm install @ngx-translate/core@^9.x --save
npm install @ngx-translate/http-loader
```

STEP 2: สร้างไฟล์ json สำหรับภาษาต่างๆ ใน assets/i18n/

เช่นต้องการภาษาไทย กับอังกฤษ ก็สร้างไฟล์ th.json กับ en.json

th.json

```
{
  "tab_menu": {
    "home": "หน้าแรก",
    "team": "ทีมงาน",
    "finance": "การเงิน",
    "notification": "แจ้งเตือน",
    "more": "เพิ่มเติม"
  },
  "more_menu": {
    "edit": "แก้ไข",
    "setting": "ตั้งค่าระบบ",
    "points": "เดิมสะสม",
    "products": "สินค้า",
    "donate": "บริจาด",
    "question": "คำถามที่พบบ่อย",
    "policy": "นโยบายและเงื่อนไข",
    "contact": "ติดต่อระบบ",
    "logout": "ออกจากระบบ"
  }
}
```

en.json

```
{
  "tab_menu": {
```

```

    "home": "Home",
    "team": "Team",
    "finance": "Finance",
    "notification": "Notification",
    "more": "More"
  },
  "more_menu": {
    "edit": "Edit",
    "setting": "Setting",
    "points": "Point",
    "products": "Products",
    "donate": "Donate",
    "question": "Question",
    "policy": "Policy",
    "contact": "Contact us",
    "logout": "Log out"
  }
}

```

STEP 3: เพิ่มคำสั่งการ import ในไฟล์ src/app/app.module.ts

```

// Multiple Language
import { HttpClientModule, HttpClient } from '@angular/common/http';
import { TranslateModule, TranslateLoader } from '@ngx-translate/core';
import { TranslateHttpLoader } from '@ngx-translate/http-loader';
export function createTranslateLoader(http: HttpClient) {
  return new TranslateHttpLoader(http, './assets/i18n/', '.json');
}

imports: [
  BrowserModule,
  IonicModule.forRoot(MyApp),
  IonicStorageModule.forRoot(),
  HttpClientModule,
  TranslateModule.forRoot({
    loader: {
      provide: TranslateLoader,
      useFactory: createTranslateLoader,

```

```

    deps: [HttpClient]
  }
})
],

```

STEP 4: เพิ่มคำสั่ง import ในหน้าที่ต้องการเปลี่ยนภาษา เช่น src /pages/tab-more.module.ts

```

import { NgModule } from '@angular/core';
import { IonicPageModule } from 'ionic-angular';
import { TabMorePage } from './tab-more';

import { HttpClient } from '@angular/common/http';
import { TranslateModule, TranslateLoader } from '@ngx-translate/core';
import { TranslateHttpLoader } from '@ngx-translate/http-loader';
export function createTranslateLoader(http: HttpClient) {
  return new TranslateHttpLoader(http, './assets/i18n/', '.json');
}

@NgModule({
  declarations: [
    TabMorePage,
  ],
  imports: [
    IonicPageModule.forChild(TabMorePage),
    TranslateModule.forChild({
      loader: {
        provide: TranslateLoader,
        useFactory: (createTranslateLoader),
        deps: [HttpClient]
      }
    })
  ],
  exports: [
    TabMorePage
  ]
})
export class TabMorePageModule {}

```

STEP 5: เพิ่มคำสั่ง import ในหน้าที่ต้องการเปลี่ยนภาษา เช่น src/pages/tab-more.ts

```
import { TranslateService } from '@ngx-translate/core';

export class TabMorePage {

  lang: any;

  constructor(
    public navCtrl: NavController,
    public navParams: NavParams,
    public app: App,
    public storage: Storage,
    public translate: TranslateService) {

    this.lang = 'th';
    this.translate.setDefaultLang('th');
    this.translate.use('th');

  }

}
```

```
switchLanguage() {
  this.translate.use(this.lang);
  // ต้องเก็บภาษาที่ผู้ใช้เปลี่ยนลง Local Storage เพื่อไว้กำหนดภาษาเริ่มต้น
}
```

STEP 6: เพิ่มเมนู และอีเวนต์ในหน้า html เช่น src/pages/tab-more.html

```
<ion-list>
  <ion-item>
    <ion-label>Language</ion-label>
    <ion-select [(ngModel)]="lang" (ionChange)="switchLanguage()">
      <ion-option value="en">English</ion-option>
      <ion-option value="th">Thai</ion-option>
    </ion-select>
  </ion-item>
</ion-list>
```

```
<button ion-item class="btn-list-more" (press)="doSomething()">
  <ion-icon name="ios-settings-outline" item-start></ion-icon>
  {{'more_menu.setting' | translate}}
  <ion-icon name="ios-arrow-forward-outline" item-end></ion-icon>
</button>
```

STEP 7: การเปลี่ยนภาษาในเมนู Tabs

แก้ไขในไฟล์ html เช่น src/pages/tabs.html

```
<ion-tabs>
  <ion-tab tabIcon="ios-home-outline" [root]="tab1Root" tabTitle="{{'tab_menu.home' | translate}} "></ion-tab>
  <ion-tab tabIcon="ios-people-outline" [root]="tab2Root" tabTitle="{{'tab_menu.team' | translate}} " (ionSelect)="check_tab_select()"></ion-tab>
  <ion-tab tabIcon="ios-cash-outline" [root]="tab3Root" tabTitle="{{'tab_menu.finance' | translate}} " (ionSelect)="check_tab_select()"></ion-tab>
  <ion-tab tabIcon="ios-notifications-outline" [root]="tab4Root" tabTitle="{{'tab_menu.notification' | translate}} " tabBadge="14" tabBadgeStyle="danger"
  (ionSelect)="check_tab_select()"></ion-tab>
  <ion-tab tabIcon="ios-menu-outline" [root]="tab5Root" tabTitle="{{'tab_menu.more' | translate}} " (ionSelect)="check_tab_select()"></ion-tab>
</ion-tabs>
```

** Trick สำคัญ tabTitle="{{'tab_menu.home' | translate}} " สังเกตต้องมีการเว้นช่องว่างด้านล่าง } } ด้วยอย่างน้อย 1 space
ถึงจะทำได้งาน

Module 12: การใช้งาน Google Map ใน Ionic

Step 1: Get Google Map API Key

<https://developers.google.com/maps/documentation/javascript/get-api-key>

Step 2: Add the API key to your application

```
<script async defer src="https://maps.googleapis.com/maps/api/js?key=AlzaSyBWRTeDq-  
18tZiK2lh_QCDk9WpPuI4AwW4&callback=initMap"  
type="text/javascript"></script>
```

Step 3: Add the JavaScript SDK

src/index.html

```
<script async defer src="https://maps.googleapis.com/maps/api/js?key=AlzaSyBWRTeDq-  
18tZiK2lh_QCDk9WpPuI4AwW4&callback=initMap"  
type="text/javascript"></script>
```

Step 4: Load the Map

pages/viewmap/viewmap.html

```
<ion-header>  
<ion-navbar>  
<ion-title>  
    Map  
</ion-title>  
<ion-buttons end>  
    <button ion-button (click)="addMarker()"><ion-icon name="add"></ion-icon>Add Marker</button>  
</ion-buttons>  
</ion-navbar>  
</ion-header>  
<ion-content>  
<div #map id="map"></div>
```

```
</ion-content>
```

Step 5: Load the Map

pages/viewmap/viewmap.ts

```
-----  
import { Component, ViewChild, ElementRef } from '@angular/core';  
import { NavController } from 'ionic-angular';  
declare var google;  
  
export class ViewmapPage{  
  @ViewChild('map') mapElement: ElementRef;  
  map: any;  
  constructor(public navCtrl: NavController) {  
  }  
  ionViewDidLoad(){  
    this.loadMap();  
  }  
  
  loadMap(){  
    let latLng = new google.maps.LatLng(-34.9290, 138.6010);  
    let mapOptions = {  
      center: latLng,  
      zoom: 15,  
      mapTypeId: google.maps.MapTypeId.ROADMAP  
    }  
    this.map = new google.maps.Map(this.mapElement.nativeElement, mapOptions);  
  }  
}
```

Step 6: Style the Map

pages/viewmap/viewmap.scss

```
-----  
.ios, .md {  
  page-viewmap{  
    .scroll {  
      height: 100%  
    }  
    #map {  
      width: 100%;  
      height: 100%;  
    }  
  }  
}  
}
```

Step 7: Add the Geolocation plugin

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

Step 8: Add Geolocation provider to app.module.ts

src/app/app.module.ts

```
-----  
import { Geolocation } from '@ionic-native/geolocation';  
  
providers: [  
  StatusBar,  
  SplashScreen,  
  Geolocation,  
  {provide: ErrorHandler, useClass: IonicErrorHandler},  
]
```

Step 9: Modify location.ts

```
src/pages/viewmap/viewmap.ts
```

```
-----  
import { Geolocation } from '@ionic-native/geolocation';  
  
constructor(public navCtrl: NavController, public geolocation: Geolocation) {  
  
}  
  
loadMap(){  
  this.geolocation.getCurrentPosition().then((position) => {  
    let latLng = new google.maps.LatLng(position.coords.latitude, position.coords.longitude);  
    let mapOptions = {  
      center: latLng,  
      zoom: 15,  
      mapTypeId: google.maps.MapTypeId.ROADMAP  
    }  
    this.map = new google.maps.Map(this.mapElement.nativeElement, mapOptions);  
  }, (err) => {  
    console.log(err);  
  });  
}
```

Step 10: Adding Markers and Info Windows to the Map

```
pages/viewmap/viewmap.ts
```

```
-----  
addMarker() {  
  
  let marker = new google.maps.Marker({  
    map: this.map,  
    animation: google.maps.Animation.DROP,  
    position: this.map.getCenter()  
}
```

```

});
```

```

let content = "<h4>Information!</h4>";
this.addInfoWindow(marker, content);
}
addInfoWindow(marker, content) {
let infoWindow = new google.maps.InfoWindow({
  content: content
});
google.maps.event.addListener(marker, 'click', () => {
  infoWindow.open(this.map, marker);
});
}

```

and ...Add marker when map start

```
----
```

```

loadMap() {
  var im = 'https://elanmakeup.com/wp-content/uploads/sites/46/2017/10/location-icon.png';

  this.geolocation.getCurrentPosition().then((position) => {

    let latLng = new google.maps.LatLng(position.coords.latitude, position.coords.longitude);

    let mapOptions = {
      center: latLng,
      zoom: 15,
      mapTypeId: google.maps.MapTypeId.ROADMAP
    }

    this.map = new google.maps.Map(this.mapElement.nativeElement, mapOptions);
  })
}
```

```
new google.maps.Marker({  
  position: latLng,  
  map: this.map,  
  icon: im  
});  
, (err) => {  
  console.log(err);  
});  
}
```

Module 13: Workshop การถ่ายภาพหรือเลือกจาก gallery และอัปโหลดขึ้น Server

Step 1:

```
-----  
npm install @ionic-native/camera --save  
npm install @ionic-native/file --save  
npm install @ionic-native/file-path --save  
npm install @ionic-native/transfer --save
```

Step 2:

```
-----  
ionic cordova plugin add cordova-plugin-camera --save  
ionic cordova plugin add cordova-plugin-file --save  
ionic cordova plugin add cordova-plugin-filpath --save  
ionic cordova plugin add cordova-plugin-file-transfer --save
```

Step 3: app.module.ts

```
-----  
import { File } from '@ionic-native/file/ngx';  
import { Transfer } from '@ionic-native/transfer';  
import { FilePath } from '@ionic-native/file-path/ngx';  
import { Camera } from '@ionic-native/camera';  
  
providers: [  
  StatusBar,  
  SplashScreen,  
  File,  
  Transfer,  
  Camera,  
  FilePath,  
  {provide: ErrorHandler, useClass: IonicErrorHandler}]
```

Step 4: pages/home/home.html

```
-----  
<ion-header>  
  <ion-navbar>  
    <ion-title>Image</ion-title>  
  </ion-navbar>  
</ion-header>  
  
<ion-content>  
  
<ion-grid>  
  <ion-row>  
    <ion-col col-12>  
        
    </ion-col>  
  </ion-row>  
</ion-grid>  
</ion-content>  
  
<ion-footer padding text-center>  
  
  <button ion-button icon-left (click)="presentActionSheet()">  
    <ion-icon name="camera"></ion-icon>Select Image  
  </button>  
  
  <button ion-button icon-left (click)="uploadImage()" [disabled]="lastImage === null">  
    <ion-icon name="cloud-upload"></ion-icon>Upload  
  </button>  
</ion-footer>
```

Step 5: Building the Ionic images capturing class

```
src/pages/home/home.ts
```

```
-----  
import { NavController, ActionSheetController, ToastController, Platform, LoadingController, Loading }  
from 'ionic-angular';  
  
import { File } from '@ionic-native/file';  
import { Transfer, TransferObject } from '@ionic-native/transfer';  
import { FilePath } from '@ionic-native/file-path';  
import { Camera } from '@ionic-native/camera';  
  
declare var cordova: any;  
  
@Component({  
  selector: 'page-home',  
  templateUrl: 'home.html'  
})  
  
export class HomePage {  
  lastImage: string = null;  
  loading: Loading;  
  constructor(public navCtrl: NavController, private camera: Camera, private transfer: Transfer,  
  private file: File, private filePath: FilePath, public actionSheetCtrl: ActionSheetController, public  
  toastCtrl: ToastController, public platform: Platform, public loadingCtrl: LoadingController) {}  
  
  public presentActionSheet() {  
    let actionSheet = this.actionSheetCtrl.create({  
      title: 'Select Image Source',  
      buttons: [  
        {  
          text: 'Load from Library',  
          handler: () => {  
            this.takePicture(this.camera.PHOTOLIBRARY);  
          }  
        }  
      ]  
    });  
    actionSheet.present();  
  }  
  takePicture(sourceType) {  
    const options = {  
      quality: 100,  
      destinationType: this.camera.DestinationType.DATA_URL,  
      encodingType: this.camera.EncodingType.JPEG,  
      targetWidth: 800,  
      targetHeight: 600,  
      sourceType: sourceType  
    };  
    this.camera.getPicture(options).then((imageData) => {  
      // Handle imageData  
    }, (err) => {  
      console.log(err);  
    });  
  }  
}
```

```

    },
    {
      text: 'Use Camera',
      handler: () => {
        this.takePicture(this.camera.PictureSourceType.CAMERA);
      }
    },
    {
      text: 'Cancel',
      role: 'cancel'
    }
  ]
});

actionSheet.present();
}

}

```

Step 6: src/pages/home/home.ts

```

-----
public takePicture(sourceType) {
  // Create options for the Camera Dialog
  var options = {
    quality: 100,
    sourceType: sourceType,
    saveToPhotoAlbum: false,
    correctOrientation: true
  };

  // Get the data of an image
  this.camera.getPicture(options).then((imagePath) => {
    // Special handling for Android library
    if (this.platform.is('android') && sourceType === this.camera.PictureSourceType.PHOTOLIBRARY) {

```

```

this.filePath.resolveNativePath(imagePath)
.then(filePath => {
  let correctPath = filePath.substr(0, filePath.lastIndexOf('/') + 1);
  let currentName = imagePath.substring(imagePath.lastIndexOf('/') + 1,
imagePath.lastIndexOf('?'));
  this.copyFileToLocalDir(correctPath, currentName, this.createFileName());
});
} else {
  var currentName = imagePath.substr(imagePath.lastIndexOf('/') + 1);
  var correctPath = imagePath.substr(0, imagePath.lastIndexOf('/') + 1);
  this.copyFileToLocalDir(correctPath, currentName, this.createFileName());
}
}, (err) => {
  this.presentToast('Error while selecting image.');
});
}

```

Step 7: src/pages/home/home.ts

```
-----
// Create a new name for the image
private createFileName() {
  var d = new Date(),
  n = d.getTime(),
  newFileName = n + ".jpg";
  return newFileName;
}
```

Step 8: src/pages/home/home.ts

```
// Copy the image to a local folder
private copyFileToLocalDir(namePath, currentName, newFileName) {
  this.file.writeFile(namePath, currentName, cordova.file.dataDirectory, newFileName).then(success =>
{
  this.lastImage = newFileName;
}, error => {
```

```
this.presentToast('Error while storing file.');

});

}
```

Step 9: src/pages/home/home.ts

```
-----
private presentToast(text) {
  let toast = this.toastCtrl.create({
    message: text,
    duration: 3000,
    position: 'top'
  });
  toast.present();
}
```

Step 10: src/pages/home/home.ts

```
-----
// Always get the accurate path to your apps folder
public pathForImage(img) {
  if (img === null) {
    return '';
  } else {
    return cordova.file.dataDirectory + img;
  }
}
```

Step 11: src/pages/home/home.ts

```
-----  
public uploadImage() {  
    // Destination URL  
    var url = "http://localhost/uploadimageapi/upload.php";  
  
    // File for Upload  
    var targetPath = this.pathForImage(this.lastImage);  
  
    // File name only  
    var filename = this.lastImage;  
  
    var options = {  
        fileKey: "file",  
        fileName: filename,  
        chunkedMode: false,  
        mimeType: "multipart/form-data",  
        params : {'fileName': filename}  
    };  
  
    const fileTransfer: TransferObject = this.transfer.create();  
  
this.loading = this.loadingCtrl.create({  
    content: 'Uploading...',  
});  
this.loading.present();  
  
    // Use the FileTransfer to upload the image  
    fileTransfer.upload(targetPath, url, options).then(data => {  
        this.loading.dismissAll()  
        this.presentToast('Image successful uploaded.');//  
    }, err => {  
        this.loading.dismissAll()  
        this.presentToast('Error while uploading file.');//  
    }  
};
```

```
});  
}
```

Step 12: C:/xampp/htdocs/uploadimageapi/upload.php

Creating our simple image upload PHP server

```
-----  
<?php  
header('Access-Control-Allow-Origin: *');  
$target_path = "uploads/";  
$target_path = $target_path . basename( $_FILES['file']['name']);  
if (move_uploaded_file($_FILES['file']['tmp_name'], $target_path)) {  
    echo "Upload and move success";  
} else {  
    echo $target_path;  
    echo "There was an error uploading the file, please try again!";  
}  
?>
```

Step 13:

```
-----  
ionic cordova platform add android
```

Step 14:

```
-----  
ionic cordova build android  
ionic cordova run android
```

Module 14: Barcode Scanner

Step 1:

```
ionic cordova plugin add phonegap-plugin-barcodescanner --save  
npm install --save @ionic-native/barcode-scanner
```

Step 2: src\app\app.module.ts

```
import { BarcodeScanner } from '@ionic-native/barcode-scanner';  
  
providers: [  
  StatusBar,  
  SplashScreen,  
  BarcodeScanner,  
  {provide: ErrorHandler, useClass: IonicErrorHandler}  
]
```

Step 3: src/pages/barcode/barcode.html

```
<ion-header>  
  <ion-navbar>  
    <ion-title>Home</ion-title>  
  </ion-navbar>  
</ion-header>  
  
<ion-content padding>  
  <ion-card>  
    <ion-card-content>  
      <ion-list>  
        <div ion-item *ngFor="let item of data">  
          <a href="{{item}}>{{item}}</a>  
        </div>  
      </ion-list>
```

```

</ion-card-content>
</ion-card>
</ion-content>
<ion-footer padding>
  <ion-toolbar position="bottom">
    <button ion-button block outline color="primary" (click)="scan()">Scan</button>
  </ion-toolbar>
</ion-footer>

```

Step 4: src/pages/barcode/barcode.ts

```

-----
import { BarcodeScanner } from '@ionic-native/barcode-scanner';

export class BarcodePage {

  data = [];

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

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


```

Step 4: ionic cordova build android

```

-----
ionic cordova run android

```