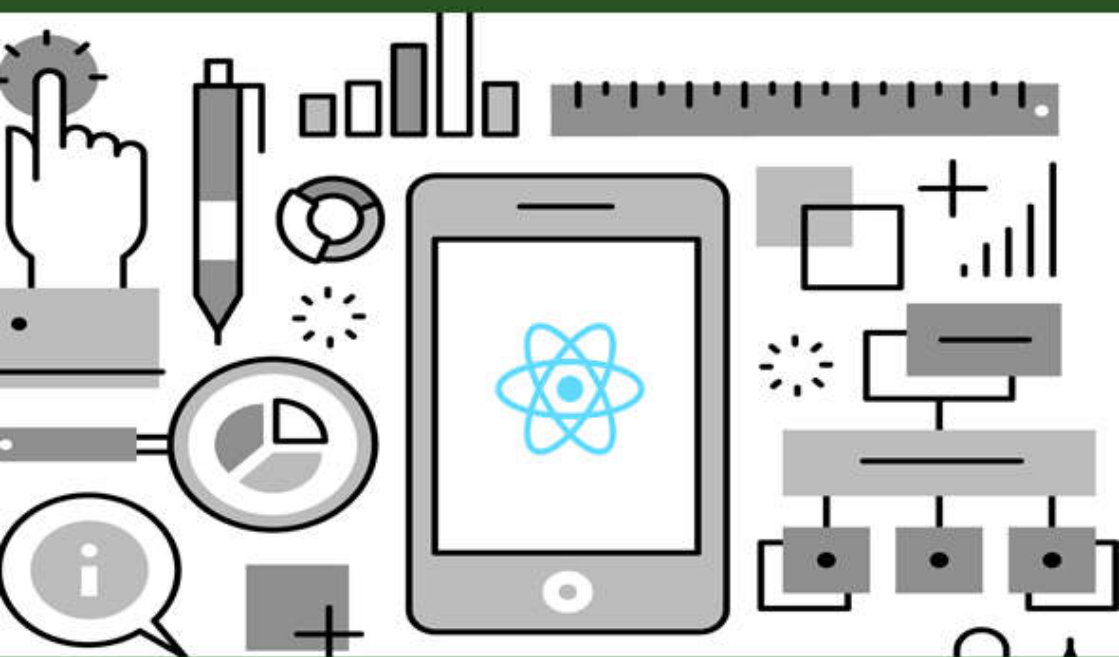


พัฒนา Mobile Application ด้วย **React Native**



โค้ชเอก

CodingThailand

พัฒนา Mobile Application ด้วย React Native

by โค้ชเอก

คู่มือเล่มนี้ผู้เขียนตั้งใจจัดทำขึ้นเพื่อใช้เป็นแนวทางในการเริ่มต้นใช้งาน React Native ซึ่งจะเน้นเนื้อหาเกี่ยวกับการเตรียมสภาพแวดล้อมต่างๆ และการติดตั้ง React Native หวังว่าคู่มือเล่มนี้จะเป็นประโยชน์ และประหยัดเวลาการเรียนรู้ของทุกคนครับ

“จงเอาชนะความไม่รู้ ด้วยการพัฒนาตัวเอง และลงมือทำอย่างสม่ำเสมอ”



©2019 CodingThailand.com

สารบัญ

React Native พื้นฐาน 4

- React Native คืออะไร
- ความรู้พื้นฐานที่ต้องมี
- ภาพใหญ่ของ React Native กับการติดต่อกับฐานข้อมูล

การเตรียมตัวก่อนติดตั้ง React Native 6

- ก่อนใช้งาน React Native ต้องติดตั้งอะไรบ้าง
- ตรวจสอบโปรแกรมต่างๆ
- การติดตั้ง Android Studio
- ขั้นตอนการสร้าง Emulator

การติดตั้ง React Native 20

- การติดตั้ง React Native CLI
- การรันโปรเจกเพื่อทดสอบ App
- ขั้นตอนการรันโปรเจกโดยใช้ Emulator
- ขั้นตอนการรันโปรเจกบนเครื่องจริง (USB)

การใช้งาน Visual Studio Code	28
- การติดตั้ง Extensions สำหรับ React Native	
- ทดลองเขียน React Native	
 สรุปเทคนิค และคำสั่งที่ใช้บ่อย	 33
 การใช้งาน React Navigation	 34
- การติดตั้ง React Navigation	
- ทดลองสร้างเพจใหม่ด้วย React Navigation	

React Native พื้นฐาน

React Native คืออะไร

React Native เป็น framework ยอดนิยมสำหรับสร้าง native mobile apps ด้วยภาษา JavaScript โดยใช้ React library ในการเขียนโค้ด สำหรับคนที่ต้องการศึกษา React Native ก็ควรมีพื้นฐานการเขียน React มาก่อน

Note: เว็บไซต์หลักของ React Native <https://facebook.github.io/react-native>

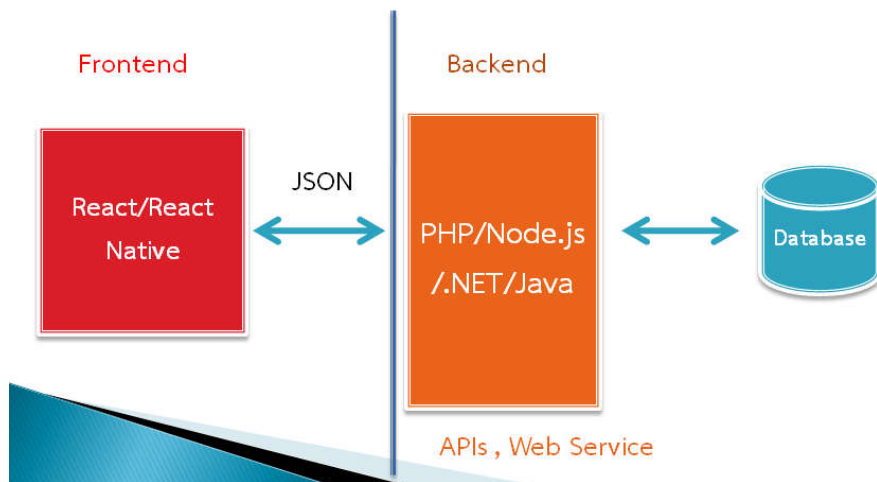
Note: เว็บไซต์หลักของ React <https://reactjs.org/>

ความรู้พื้นฐานที่ต้องมี

ก่อนเขียน React Native ควรมีพื้นฐานความรู้ ดังต่อไปนี้

1. ภาษา JavaScript ES6 ขึ้นไป
2. ต้องรู้หลักการในการเขียน React ได้แก่
 - 2.1 JSX
 - 2.2 Components และ Props
 - 2.3 State และ Lifecycle
 - 2.4 การจัดการ Events ใน React

ภาพใหญ่ของ React Native กับการติดต่อกับฐานข้อมูล



จากรูปด้านบนจะเห็นว่า React Native หากเรามองว่าเป็นส่วนของ Frontend ก็ได้ (Mobile App) หากเราต้องการติดต่อกับฐานข้อมูลก็ต้องมีการเขียน Backend อาจจะเขียนในรูปแบบของ APIs ด้วยภาษาที่เราถนัดได้เลย ไม่ว่าจะเป็น PHP, Node.js, .NET หรือ JAVA เป็นต้น โดย Backend จะส่งข้อมูลมาในรูปแบบของ JSON ไปให้กับ React Native นั่นเอง

การเตรียมตัวก่อนติดตั้ง React Native

ก่อนใช้งาน React Native ต้องติดตั้งอะไรบ้าง

ก่อนที่จะใช้งาน React Native นั้น จะต้องติดตั้งโปรแกรมต่างๆ ดังนี้

1. เตรียม Notebook หรือ Mac สำหรับ Notebook แนะนำเป็น Windows 10 แบบ 64bit จะดีมาก (หรือถ้าไม่มีอย่างน้อยต้อง Windows 7 64bit) Ram ต้องไม่น้อยกว่า 6 GB (แนะนำ 8 GB ขึ้นไป)
2. ติดตั้ง git ดาวน์โหลดได้ที่นี้ โดยคลิก next อย่างเดียวจนเสร็จเรียบร้อย
<https://git-scm.com/download>
3. ติดตั้ง Node.js เวอร์ชัน LTS โดยคลิก next อย่างเดียวจนเสร็จเรียบร้อย ตามลิงก์
<https://nodejs.org/en/>
4. ติดตั้ง Python เวอร์ชัน 2 เท่านั้น
<https://www.python.org/downloads/release/python-2715/>
5. ติดตั้ง Visual Studio Code คลิก next อย่างเดียวจนเสร็จเรียบร้อย ตามลิงก์
<https://code.visualstudio.com/>
6. ติดตั้ง JRE เวอร์ชัน 8 เท่านั้น คลิก next อย่างเดียวจนเสร็จเรียบร้อย
<https://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>
7. ติดตั้ง Java JDK เวอร์ชัน 8 คลิก next อย่างเดียวจนเสร็จเรียบร้อย
<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

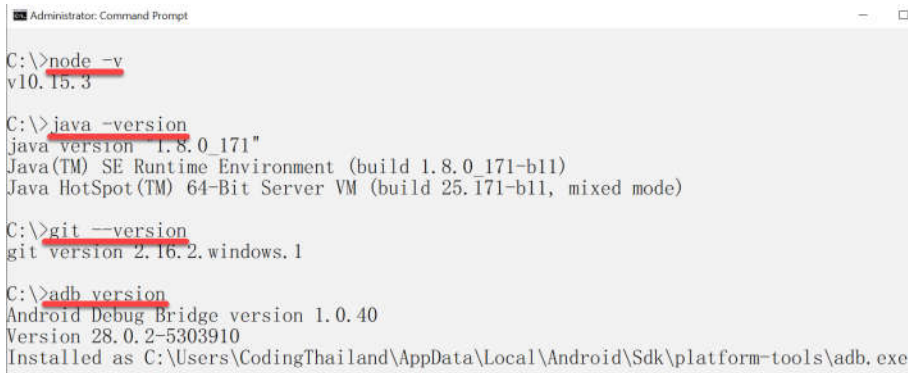
8. ติดตั้ง Android Studio (คลิก next อย่างเดียวจนเสร็จเรียบร้อย)

<https://developer.android.com/studio/>

9. เปิด Virtualization Technology (Intel VT-x) ที่ BIOS (กด F2 หรือ Delete หรือ Esc เพื่อเข้า BIOS) มองหาเมนู หรือคำว่า "VT", "Virtualization Technology", หรือ "VT-d." แล้วเปิดใช้งาน จากนั้นบันทึก แล้ว restart เครื่องอีกครั้ง

ตรวจสอบโปรแกรมต่างๆ

หลังจากที่ติดตั้งโปรแกรมต่างๆ จนครบเรียบร้อยแล้ว ก็ลองทดสอบว่าโปรแกรมที่เราได้ติดตั้งนั้น เรียบร้อยดีหรือไม่ โดยให้เปิด **Command Prompt** ขึ้นมาแล้วพิมพ์ตรวจสอบโปรแกรม ดังนี้



```
Administrator: Command Prompt
C:\>node -v
v10.15.3

C:\>java -version
java version "1.8.0_171"
Java(TM) SE Runtime Environment (build 1.8.0_171-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.171-b11, mixed mode)

C:\>git --version
git version 2.16.2.windows.1

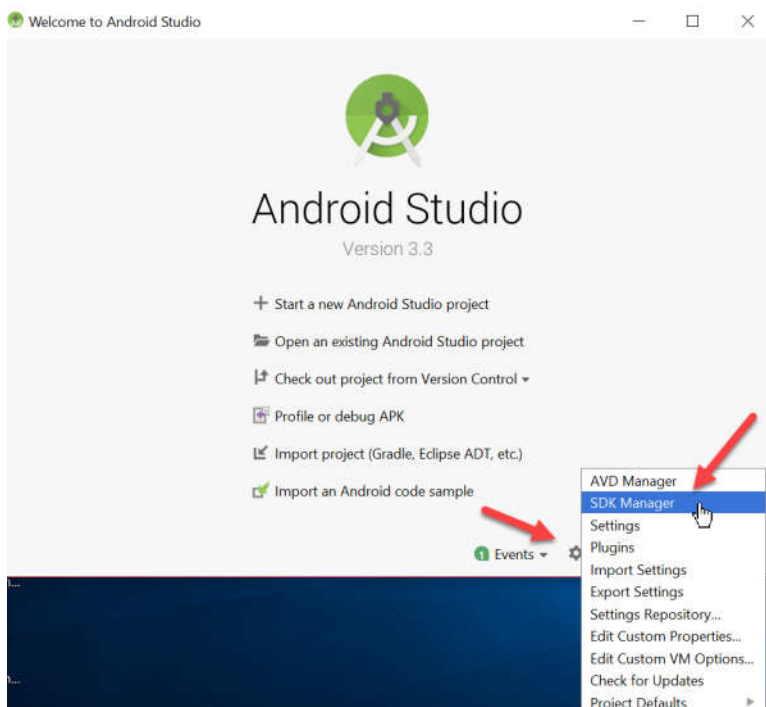
C:\>adb version
Android Debug Bridge version 1.0.40
Version 28.0.2-5303910
Installed as C:\Users\CodingThailand\AppData\Local\Android\Sdk\platform-tools\adb.exe
```

หากพิมพ์แล้วแสดงเวอร์ชันของโปรแกรม แสดงว่าการติดตั้งเรียบร้อยแล้ว

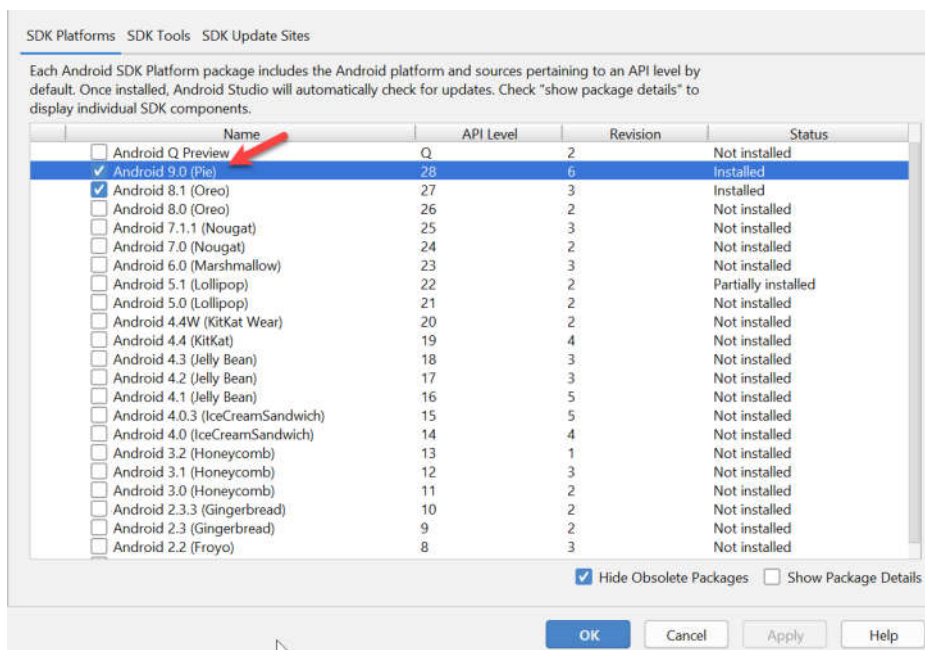
การตั้งค่า Android Studio

หลังจากติดตั้ง Android Studio เรียบร้อยแล้ว ให้ตั้งค่า และตรวจสอบการติดตั้ง SDK และ Tools ต่างๆ ดังนี้

1. เปิด Android Studio ขึ้นมา คลิกเลือกที่เมนู **Configure -> SDK Manager**

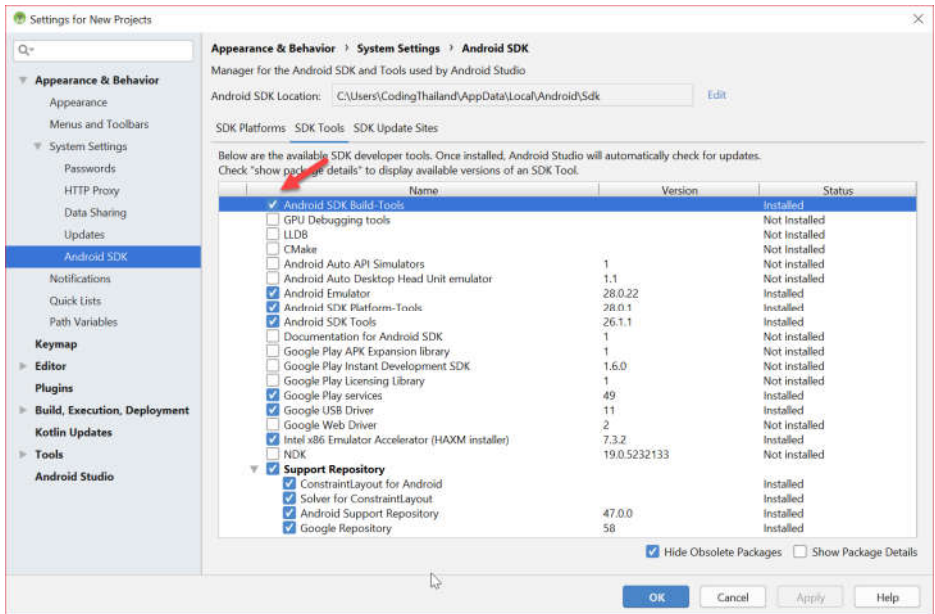


2. ตรวจสอบการติดตั้ง SDK Platforms โดยให้ดูว่ามีการเลือกที่ **Android 9 (Pie) API Level 28** แล้วหรือไม่ หากยังไม่มี ก็ให้คลิกเลือกแล้วกดปุ่ม **Apply** เพื่อติดตั้ง

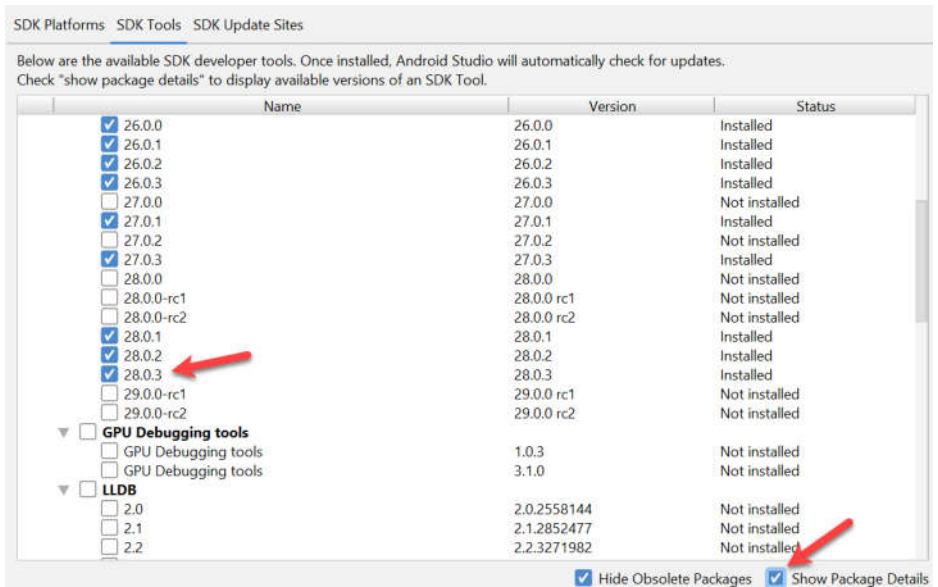


Note: ในหัวข้อนี้ ในอนาคตสามารถขยับเวอร์ชันขึ้นไปอีกได้ ดูได้จากคู่มือจากเว็บไซต์หลักของ React Native

3. คลิกที่ Tab SDK Tools ให้ตรวจสอบว่ามีตัวเลือกที่ Android SDK Build-Tools แล้ว

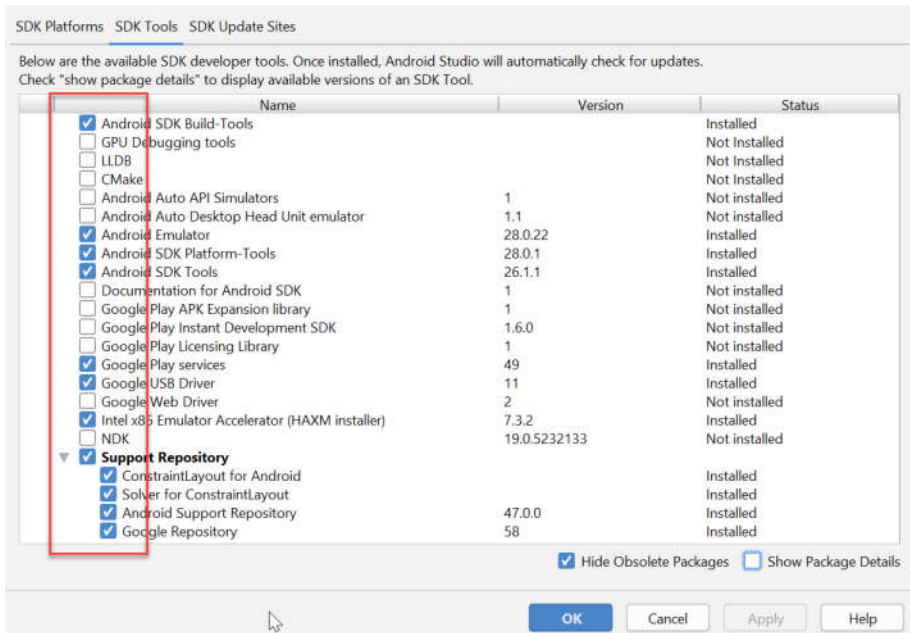


4. ให้คลิกเลือกที่ Show Package Details (มุมล่างขวา) เพื่อตรวจสอบว่าได้ติดตั้งเวอร์ชัน 28.0.3 แล้วหรือไม่ ถ้ายังไม่ได้ติดตั้ง ให้คลิกเลือก แล้วคลิกปุ่ม Apply เพื่อทำการติดตั้ง ดังรูป (ขั้นตอนนี้อาจข้ามได้ ปกติตอนรันโปรเจกจะติดตั้งให้อัตโนมัติ)



Note: ในอนาคตขั้นตอนนี้อาจมีการเปลี่ยนเลขเวอร์ชันใหม่ๆได้

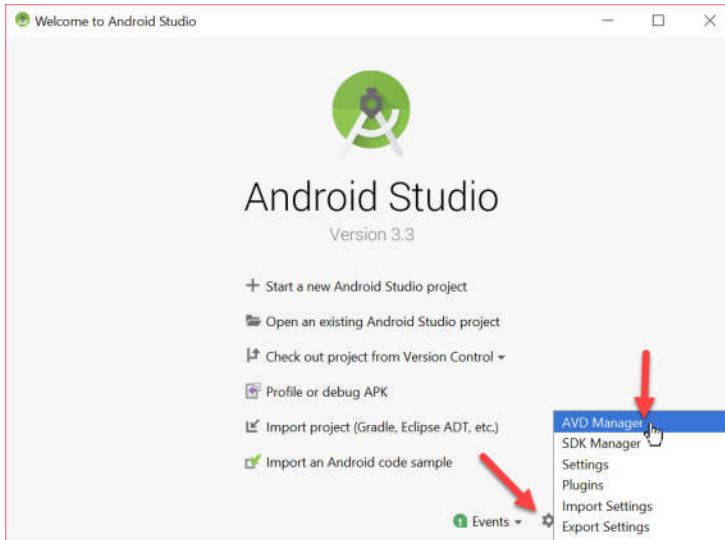
- สุดท้ายให้ตรวจสอบว่าได้ติดตั้ง หรือเลือกตามในภาพนี้ครบแล้วหรือไม่ ถ้ายังไม่ครบ ก็ให้เลือก แล้วกดปุ่ม Apply เพื่อติดตั้งให้เรียบร้อย



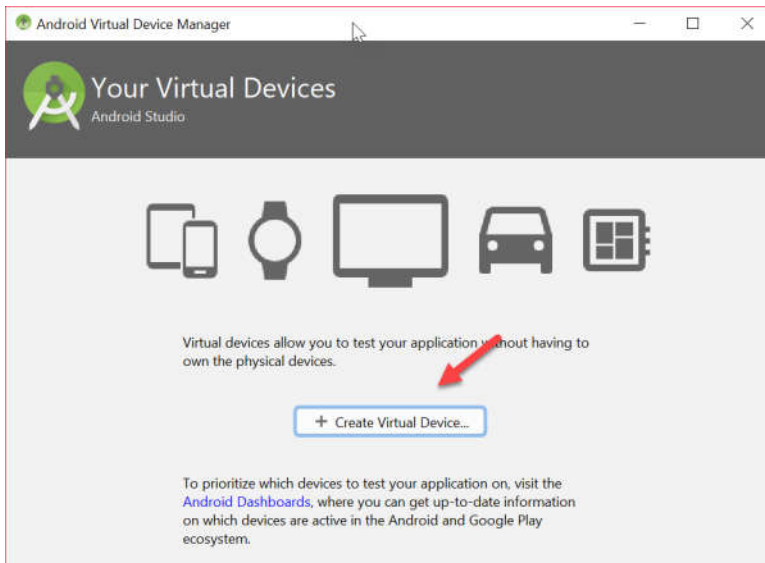
ขั้นตอนการสร้าง Emulator

เราสามารถทดสอบ Mobile App ที่เขียนได้โดยใช้เครื่องจริง (ผ่านสาย USB) หรือจะใช้ Emulator ก็ได้เช่นกัน ในหัวข้อนี้เราจะมาเรียนรู้ขั้นตอนการสร้าง Emulator ดังนี้

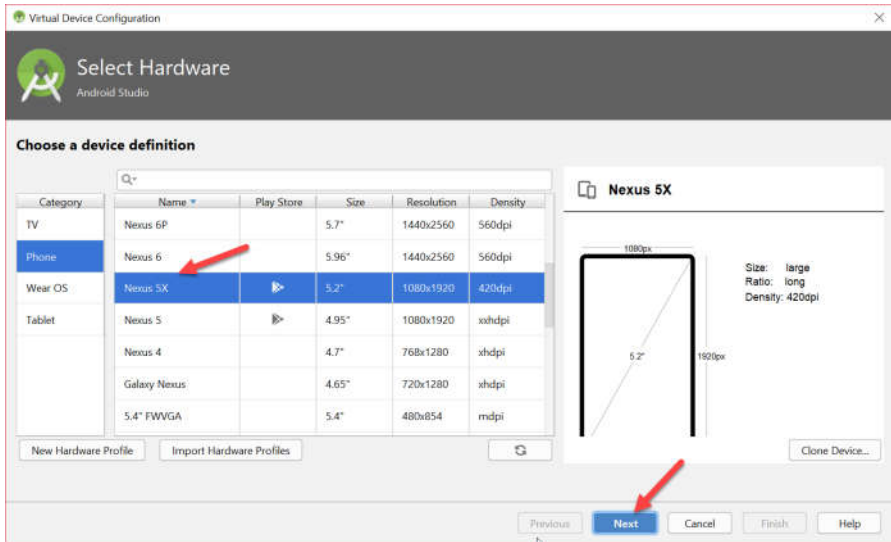
1. เปิด Android Studio ขึ้นมา คลิกเลือกที่เมนู Configure -> AVD Manager



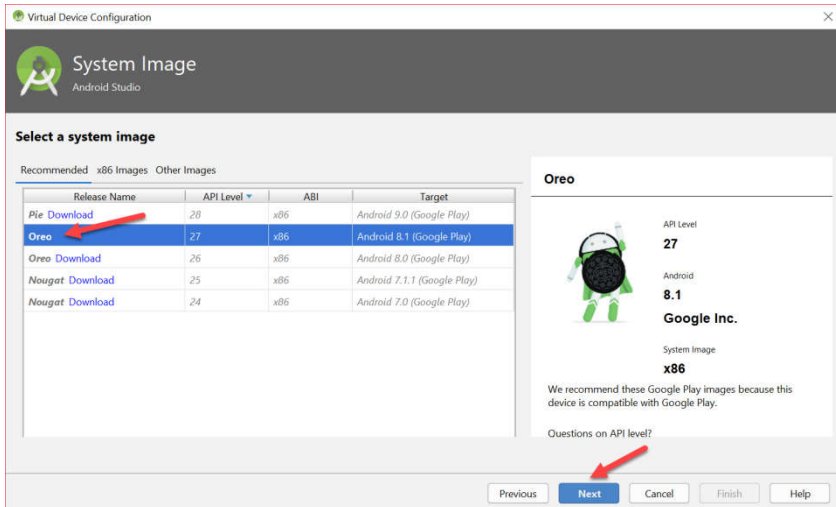
2. คลิกที่ปุ่ม Create Virtual Device...



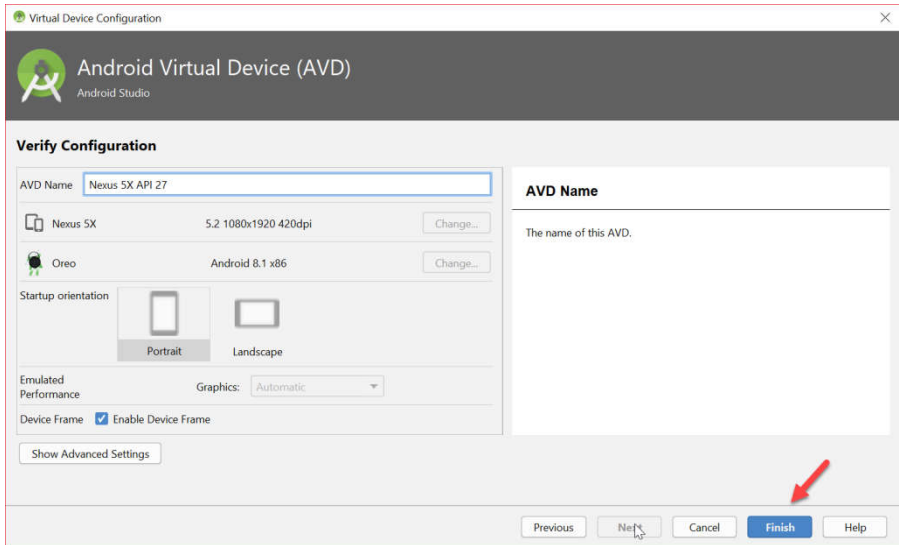
3. คลิกเลือก Nexus 5X แล้วคลิกปุ่ม Next



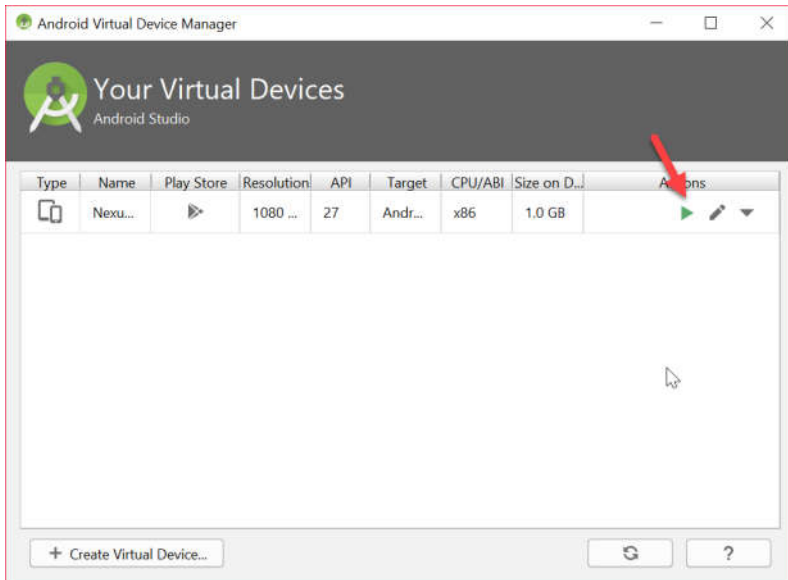
4. คลิกเลือก Oreo (API Level 27) จากนั้นคลิกปุ่ม Next (ในขั้นตอนนี้หากอยากเลือกล่าสุดก็ได้เช่นเดียวกัน)



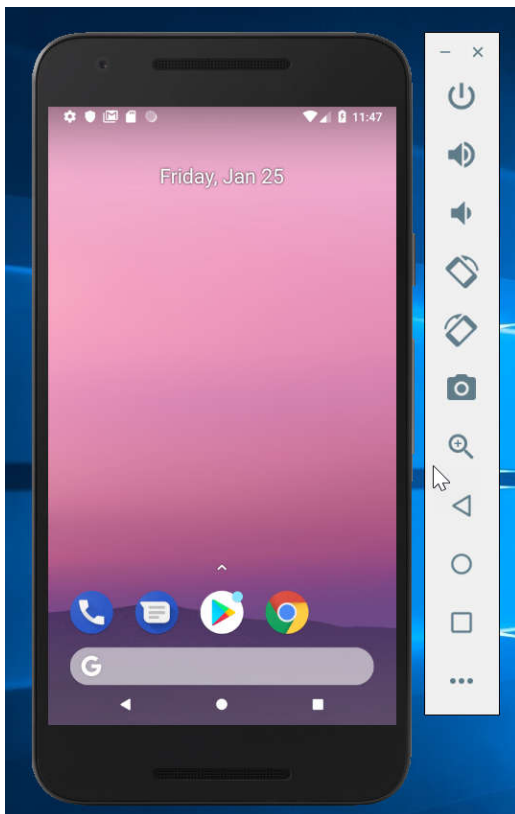
5. คลิกปุ่ม Finish



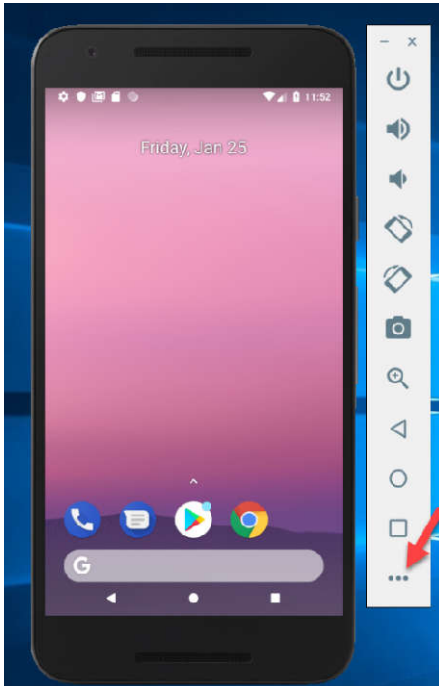
6. คลิกปุ่มสีเขียว เพื่อรัน Emulator ดังรูป



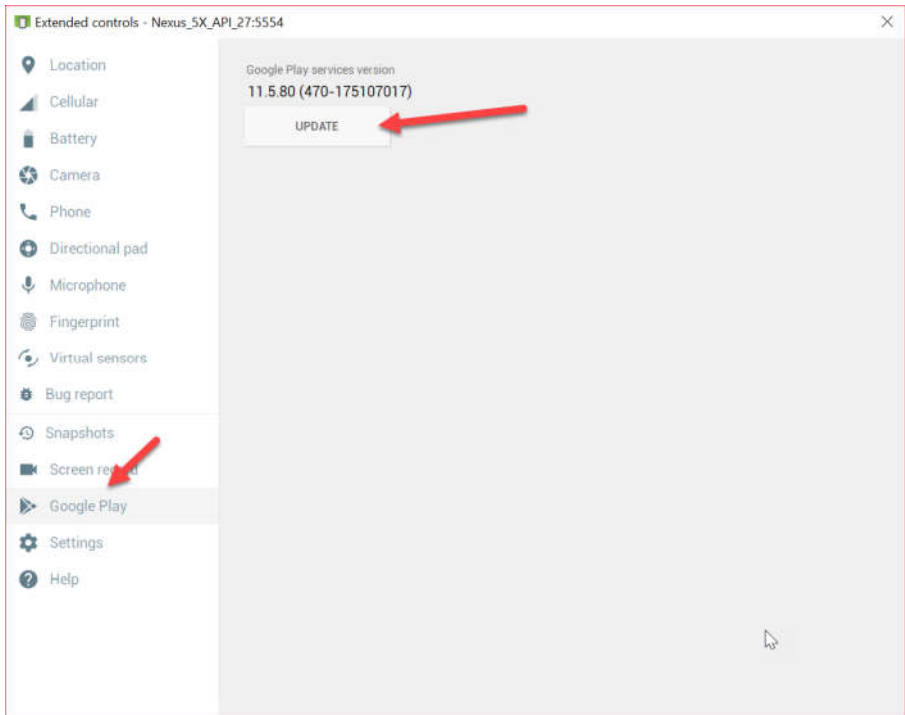
7. เมื่อกดรันแล้วจะได้ Emulator ดังรูป เป็นอันเสร็จเรียบร้อย



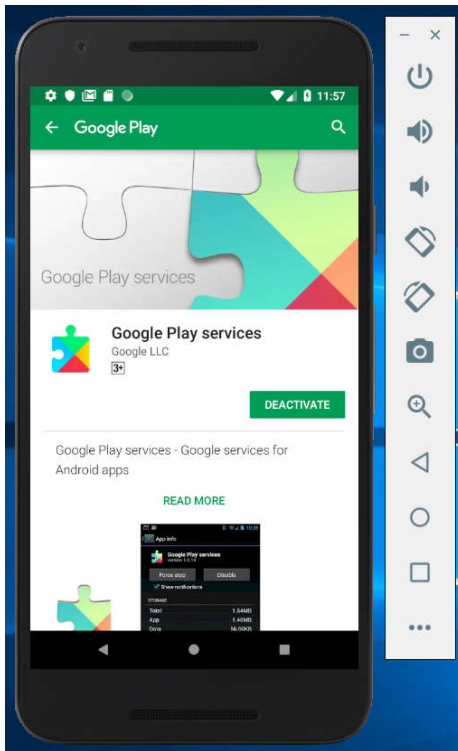
8. หลังจากได้ Emulator มาแล้ว**ควรอัปเดต Google Play Services** ให้ล่าสุดด้วย เพื่อในอนาคตเราต้องใช้ Google Maps API, Location APIs เป็นต้น ให้คลิกที่เมนู ดังรูป



9. คลิกที่เมนู Google Play (ด้านซ้าย) แล้วคลิกที่ปุ่ม UPDATE จากนั้นให้ปิดหน้าจอนี้ เพื่อกลับไปยัง Emulator



10. ให้ล็อกอินเข้าระบบของ Google Play รอสักครู่ Google Play Services จะได้รับการอัปเดตเป็นเวอร์ชันล่าสุดเรียบร้อยแล้ว



Note: เพื่อความแน่ใจสามารถไปเช็คเลขเวอร์ชันได้อีกครั้งว่าได้รับการอัปเดตหรือไม่


การติดตั้ง React Native

การติดตั้ง React Native CLI

ก่อนที่จะติดตั้ง React Native หรือสร้างโปรเจกต์ใหม่ เราจะต้องติดตั้ง React Native CLI (command line interface) ก่อน ซึ่งเป็นเครื่องมือช่วยให้เราสามารถจัดการกับโปรเจกต์ใน React Native ได้ มีขั้นตอน ดังนี้

1. ติดตั้ง React Native CLI ให้เปิด Command Prompt และพิมพ์คำสั่งดังนี้ แล้วกด Enter

```
npm install -g react-native-cli
```

 Administrator: Command Prompt

```
C:\>npm install -g react-native-cli
```

Note: ก่อนเปิด Command Prompt ให้เลือก Run as administrator ด้วยทุกครั้ง

Note: สำหรับเครื่อง Mac ต้องใส่ sudo ข้างหน้าคำสั่งด้วยหากมีปัญหาเรื่อง permission

2. สร้างโปรเจกใหม่ ด้วย คำสั่ง `react-native init <ชื่อโปรเจก>` แล้วกด Enter

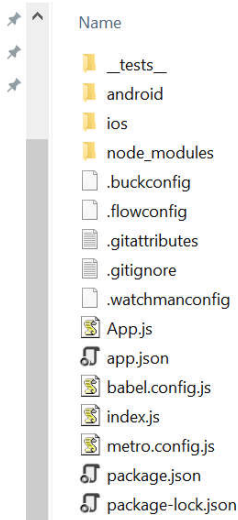
 npm

```
C:\>react-native init myrnproject
This will walk you through creating a new
'yarn' is not recognized as an internal or
operable program or batch file.
Installing react-native...
```

Note: หากใช้คำสั่งในข้อ 2 นี้ จะทำให้เราได้ React Native เวอร์ชันล่าสุด หากต้องการระบุเวอร์ชันให้ใช้คำสั่งดังนี้ `react-native init myrnproject --version 0.59.10` แล้วกด Enter

3. เมื่อติดตั้งเสร็จแล้วจะได้โฟลเดอร์ใหม่เป็นชื่อโปรเจกของเรา และไฟล์โค้ดต่างๆ จะถูกเก็บอยู่ในโฟลเดอร์นี้ ดังรูป

Windows (C:) > myrnproject >

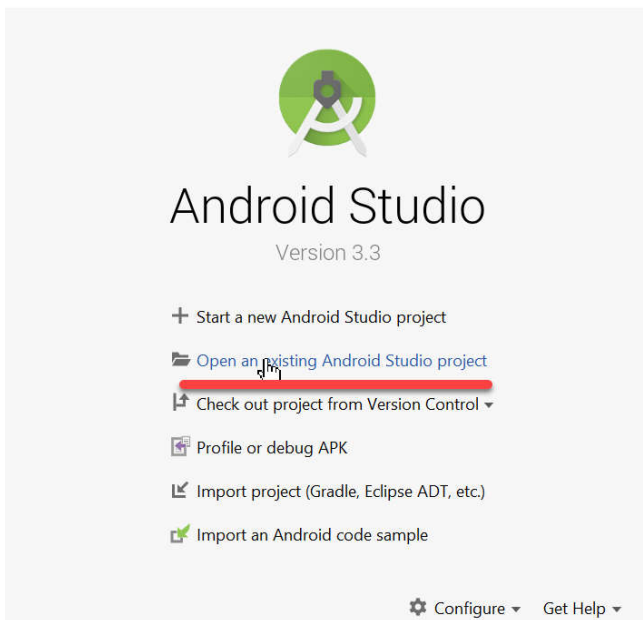


การรันโปรเจคเพื่อทดสอบ App

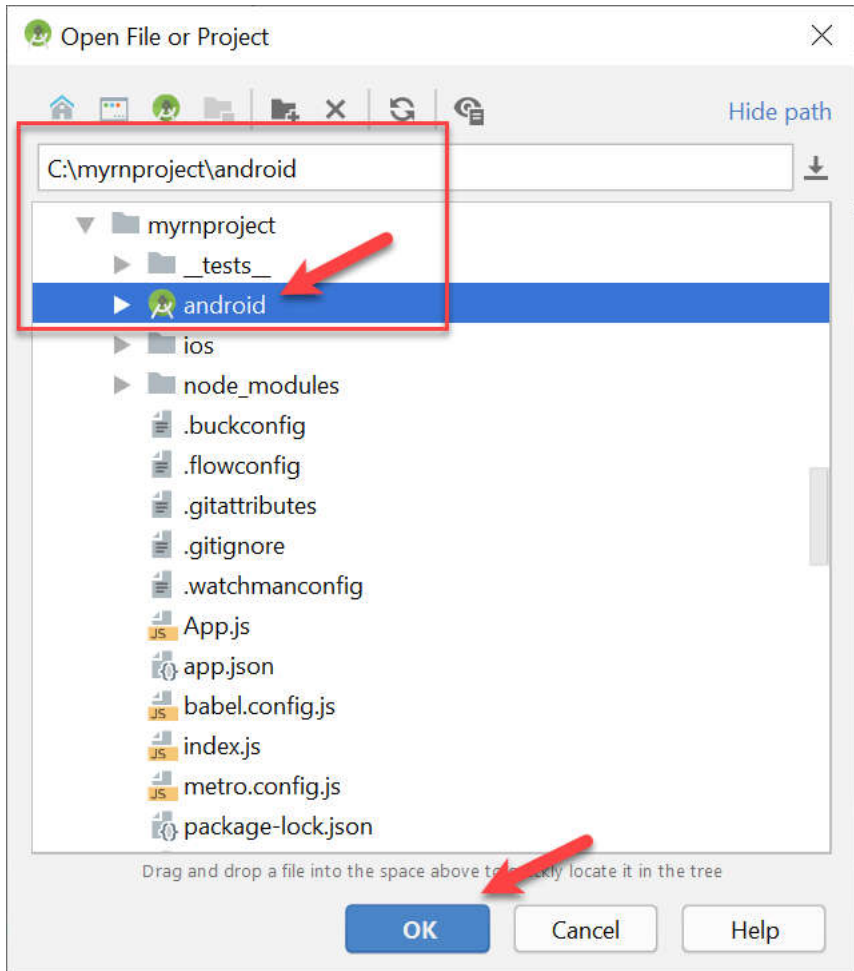
การทดสอบ App นั้นทำได้ 2 วิธีหลักๆ ด้วยกัน ได้แก่ การรันผ่าน Emulator และ การรันโดยใช้เครื่องจริง (usb)

ก่อนที่จะส่งรันโปรเจคแนะนำให้เปิด Android Studio ก่อน เพื่อ build app ให้เรียบร้อย มีขั้นตอน ดังนี้

1. เปิดโปรแกรม Android Studio ขึ้นมา คลิกเลือกที่ **Open and existing Android Studio project** ดังรูป

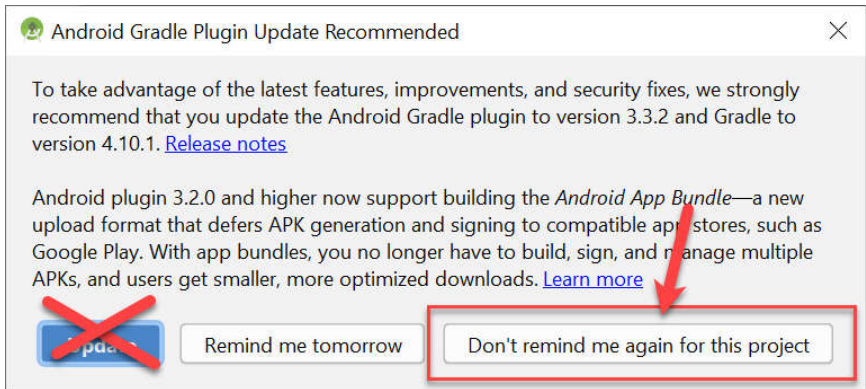


2. เลือกโฟลเดอร์โปรเจกต์ แล้วเลือกที่โฟลเดอร์ **android** จากนั้นคลิกปุ่ม OK เพื่อเปิดโปรเจกต์ขึ้นมา ดังรูป

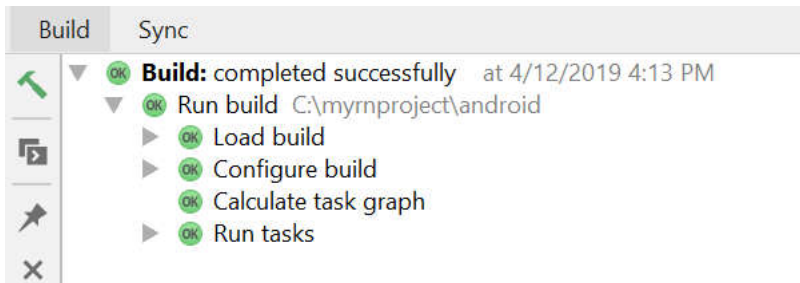


3. หากมีหน้าต่าง Android Gradle Plugin Update Recommended ให้คลิกเลือกที่ปุ่ม

Don't remind me again for this project



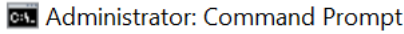
4. รอสักครู่ โปรเจค Android ของเราก็จะถูก build เรียบร้อย ดังรูป



Note: หากมี error แนะนำให้คลิกขวาแล้วเลือก run as administrator เพื่อเปิด Android Studio อีกครั้ง

ขั้นตอนการรันโปรเจคโดยใช้ Emulator

1. เปิด Command Prompt ใช้คำสั่ง `cd` เพื่อเข้าไปในโฟลเดอร์โปรเจค ดังรูป

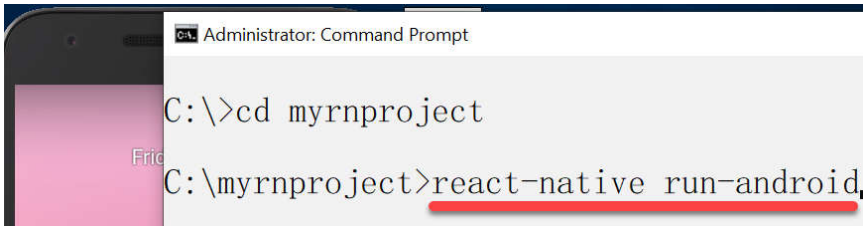


Administrator: Command Prompt

```
C:\>cd myrnproject
```

```
C:\myrnproject>
```

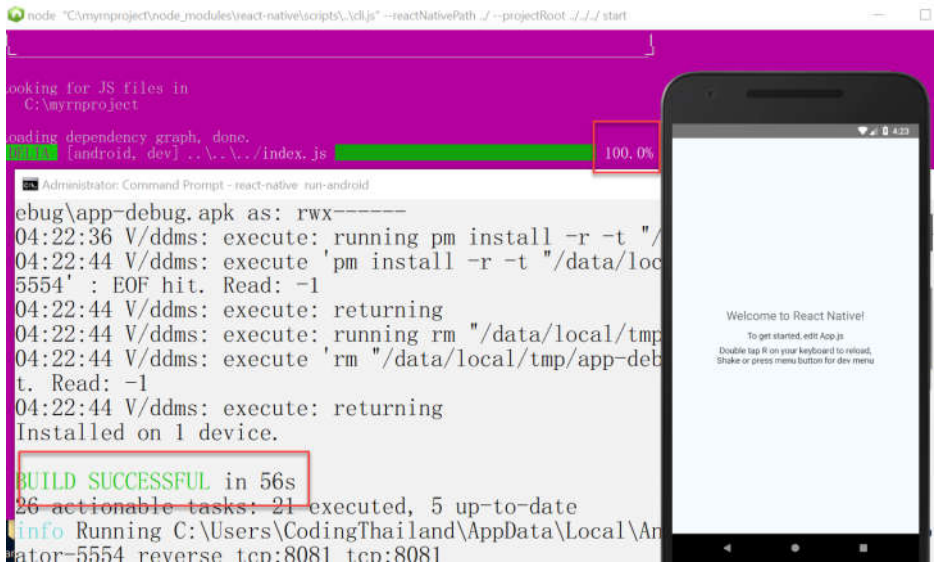
2. เปิด Emulator ค้างไว้ จากนั้นใช้คำสั่ง `react-native run-android` แล้วกด enter



Administrator: Command Prompt

```
C:\>cd myrnproject  
C:\myrnproject>react-native run-android,
```

3. รอสักครู่ โปรเจคจะถูกนำไปรันใน Emulator เรียบร้อย ดังรูป



```

node "C:\myrnproject\node_modules\react-native\scripts\cli.js" --reactNativePath ./ --projectRoot ./ ./ start
Looking for JS files in
C:\myrnproject
Building dependency graph, done.
[100.0%] (android, dev) .../index.js 100.0%
Administrator Command Prompt - react-native run-android
ebug\app-debug.apk as: rwx-----
04:22:36 V/ddms: execute: running pm install -r -t "/
04:22:44 V/ddms: execute 'pm install -r -t "/data/local/
5554' : EOF hit. Read: -1
04:22:44 V/ddms: execute: returning
04:22:44 V/ddms: execute: running rm "/data/local/tmp
04:22:44 V/ddms: execute 'rm "/data/local/tmp/app-deb
t. Read: -1
04:22:44 V/ddms: execute: returning
Installed on 1 device.

BUILD SUCCESSFUL in 56s
26 actionable tasks: 21 executed, 5 up-to-date
info Running C:\Users\CodingThailand\AppData\Local\An
rator-5554 reverse tcp:8081 tcp:8081

```

Virtual Android phone screen content:

```

Welcome to React Native

To get started, edit App.js
Double tap R on your keyboard to reload,
Shake or press menu button for dev menu

```

Note: สำหรับเครื่อง MAC ให้ใช้คำสั่ง `react-native run-ios` หรือ เปิดโปรแกรม Xcode แล้วเปิดไฟล์ในโฟลเดอร์ `ios/reactnativebkk.xcodeproj` จากนั้นก็กดปุ่มรันได้เช่นเดียวกัน

ขั้นตอนการรันโปรเจกบนเครื่องจริง (USB)

การรันบนเครื่องจริงจะต้องเตรียมสาย usb เพื่อต่อเข้ากับคอมพิวเตอร์เสียก่อน จากนั้นทำขั้นตอนดังต่อไปนี้

1. ที่โทรศัพท์เครื่องจริง ให้เปิด Debugging over USB (Developer options) โดยเข้าไปที่เมนู Settings -> About phone -> กดที่ Build number 7 ครั้ง จากนั้นกลับไปเมนู Settings -> Developer Options ให้ติ๊กเลือกที่ USB debugging
2. เมื่อเสียบสายโทรศัพท์ให้เรียบร้อย สามารถตรวจสอบว่าโทรศัพท์ได้เชื่อมต่อแล้วหรือไม่ ด้วยคำสั่ง adb devices หรือจะเปิด Android Studio แล้วกดรันตรวจการเชื่อมต่อก็ได้เช่นเดียวกัน (จะเห็นชื่ออุปกรณ์แสดงขึ้นมา)
3. จากนั้นใช้คำสั่ง react-native run-android เพื่อรันตามปกติ

Note: สำหรับ Mac ดูขั้นตอนเพิ่มเติมได้ที่ <https://facebook.github.io/react-native/docs/running-on-device>

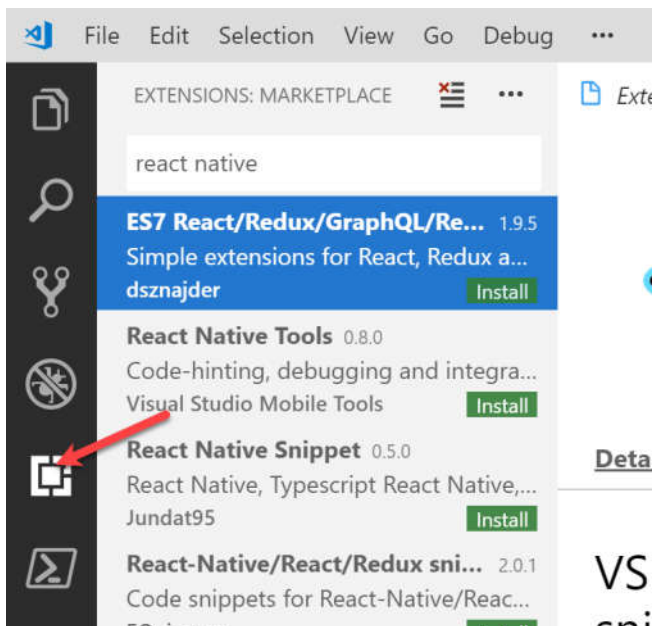
Note: ควรปิด Emulator ก่อน หากต้องการรันบนเครื่องจริง (USB)

การใช้งาน Visual Studio Code

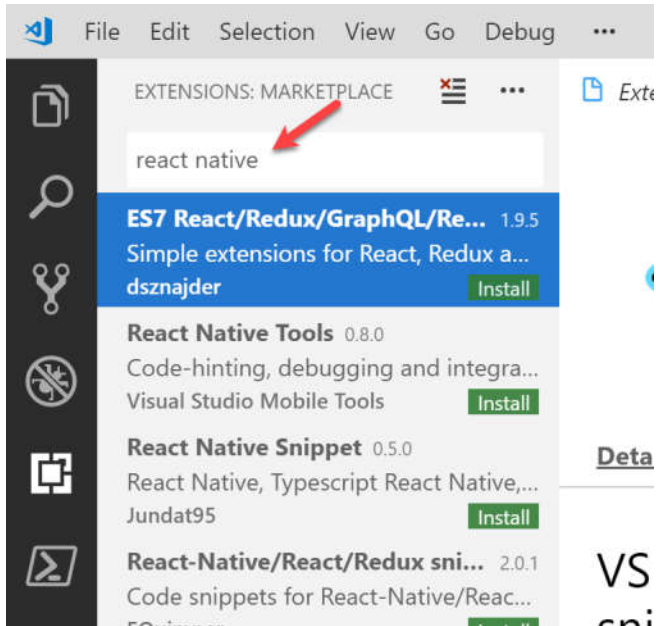
ในหัวข้อนี้เราจะติดตั้ง Extensions เพื่อช่วยให้การเขียน React Native ได้ง่ายขึ้น

การติดตั้ง Extensions สำหรับ React Native

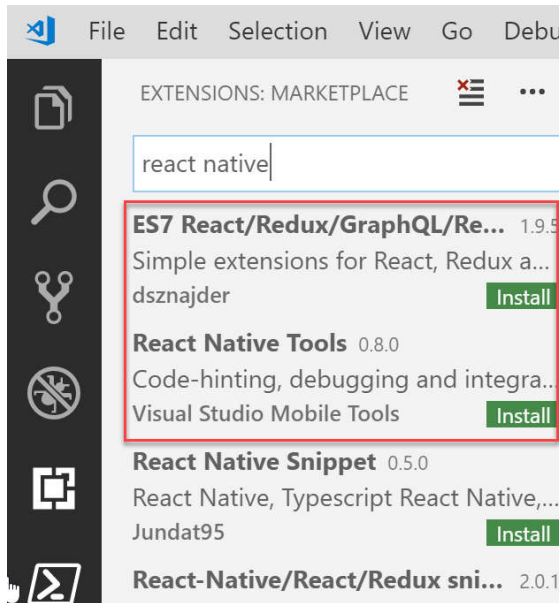
1. เปิดโปรแกรม Visual Studio Code คลิกที่ไอคอน Extensions เพื่อติดตั้ง React Native Extensions



2. พิมพ์คำว่า react native เพื่อค้นหา ดังรูป



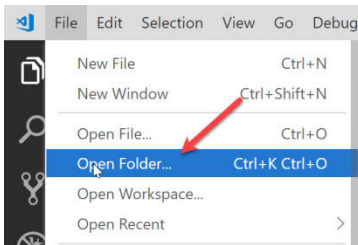
3. คลิกปุ่ม Install (สีเขียว) 2 extensions ได้แก่ ES7 React/Redux.. และ React Native Tools ดังรูป



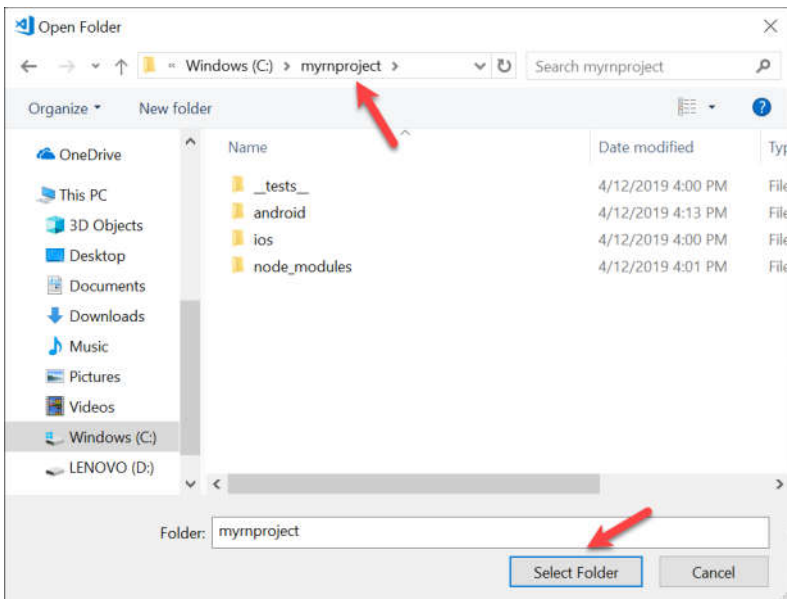
ทดลองเขียน React Native

ในหัวข้อนี้เราจะมาลองแก้ไขโค้ด เพื่อแสดง Hello World กัน

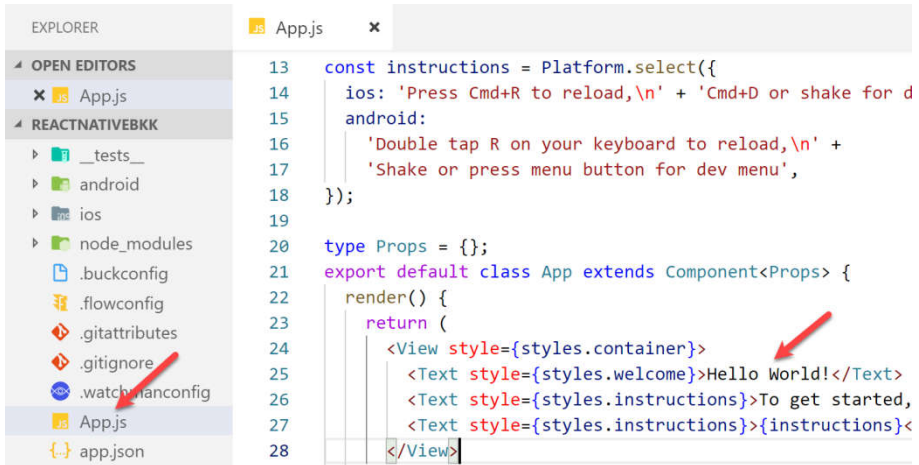
1. ใช้ Visual Studio Code เพื่อเปิดโฟลเดอร์โปรเจกต์ คลิกที่เมนู File -> Open Folder...



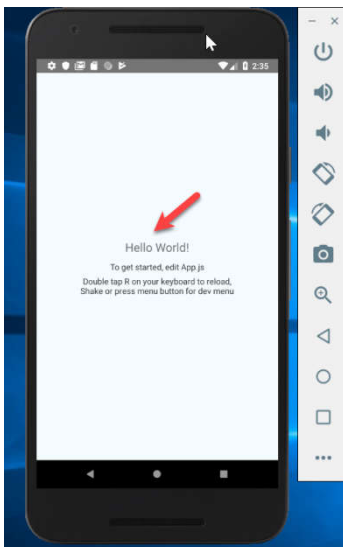
2. คลิกเลือกโฟลเดอร์โปรเจกต์



3. เปิดไฟล์ App.js แล้วเขียนโค้ด Hello World ดังรูป แล้วบันทึกไฟล์



4. หากยังไม่ได้รันโปรเจกต์ให้สั่งรันอีกครั้ง แต่ถ้ารันอยู่แล้ว สำหรับ android ให้กด R ที่คีย์บอร์ดสองครั้ง (ต่อเนื่องกัน) เพื่อ reload App สำหรับ ios ให้ Cmd+R เพื่อ reload App จะได้ผลลัพธ์ ดังรูป



สรุปเทคนิคและคำสั่งที่ใช้บ่อย

1. การสร้างโปรเจกใหม่

ใช้คำสั่ง `react-native init <ชื่อโปรเจก>` แล้วกด Enter

2. การรันโปรเจก

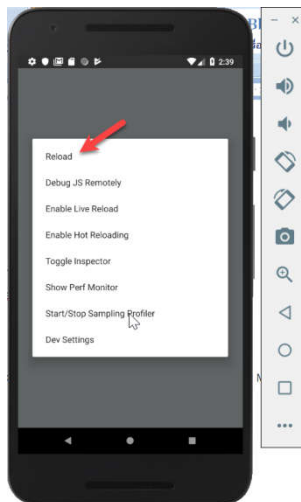
สำหรับ android ใช้คำสั่ง `react-native run-android`

สำหรับ ios ใช้คำสั่ง `react-native run-ios`

3. เมื่อแก้ไขโค้ดแล้วอยากดูผลลัพธ์

สำหรับ android ให้กด R ที่คีย์บอร์ดสองครั้งต่อเนื่องกัน หรือ กด `Ctrl + M` เพื่อเปิด Developer Menu แล้วคลิก Reload ก็ได้เช่นเดียวกัน

สำหรับ ios ให้กด `Cmd+R` เพื่อ reload หรือกด `Cmd+D` เพื่อเปิด Developer Menu



การใช้งาน React Navigation เบื้องต้น

React Navigation เป็น Library ที่สำคัญมากที่ต้องติดตั้ง มีหน้าที่ช่วยให้เรากำหนดเส้นทางให้กับเพจต่างๆ ใน App ของเรา เว็บไซต์หลัก <https://reactnavigation.org/>

การติดตั้ง React Navigation

สามารถดูขั้นตอนการติดตั้งได้ตามลิงก์นี้ <https://reactnavigation.org/docs/en/getting-started.html>

1. เปิด Command Prompt ขึ้นมา (อย่าลืม Run as administrator) พิมพ์คำสั่ง
`npm install --save react-navigation`
กด enter รอจนเสร็จเรียบร้อย
2. ติดตั้ง react-native-gesture-handler ด้วยคำสั่ง แล้วกด enter
`npm install --save react-native-gesture-handler`
3. ทำการ link native dependencies ด้วยคำสั่ง แล้วกด enter
`react-native link react-native-gesture-handler`

4. เปิดโปรแกรม Android Studio แล้วแก้ไขไฟล์ **MainActivity.java** โดยเพิ่มโค้ดในส่วนไฮไลต์สีเขียว เข้าไปดังรูป

```
package com.reactnavigation.example;

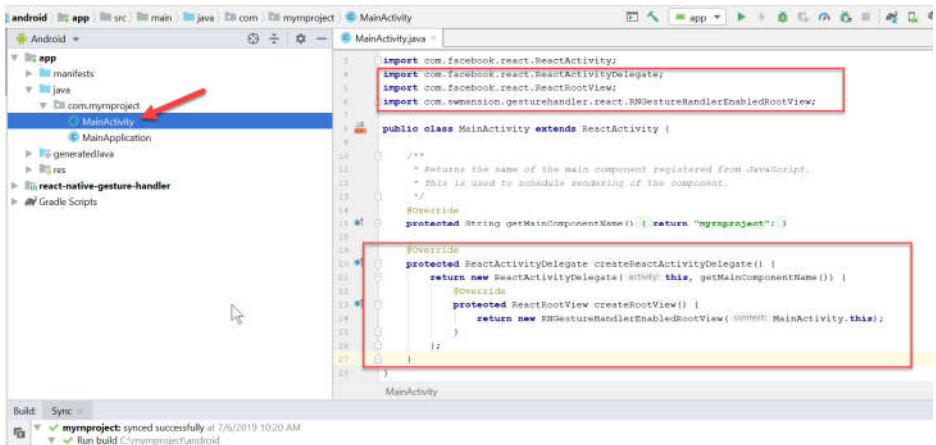
import com.facebook.react.ReactActivity;
+ import com.facebook.react.ReactActivityDelegate;
+ import com.facebook.react.ReactRootView;
+ import com.swmansion.gesturehandler.react.RNGestureHandlerEnabledRootView;

public class MainActivity extends ReactActivity {

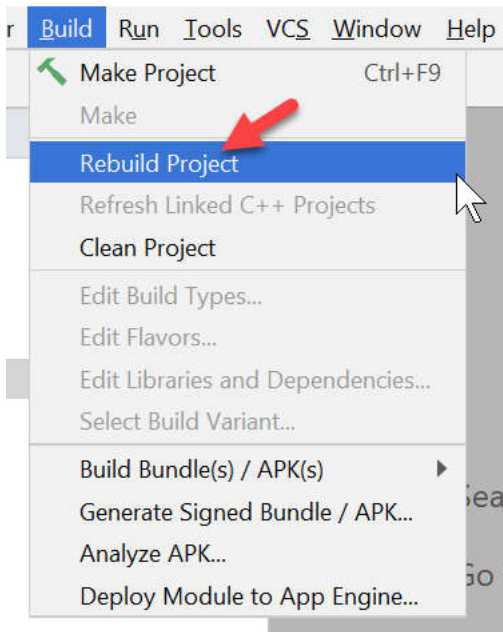
    @Override
    protected String getMainComponentName() {
        return "Example";
    }

+ @Override
+ protected ReactActivityDelegate createReactActivityDelegate() {
+     return new ReactActivityDelegate(this, getMainComponentName()) {
+         @Override
+         protected ReactRootView createRootView() {
+             return new RNGestureHandlerEnabledRootView(MainActivity.this);
+         }
+     };
+ }
}
```

แนะนำให้ Copy จากเว็บมาได้เลย อย่าลืมลบเครื่องหมายบกวาดด้านหน้าออกด้วย



5. จากนั้นสั่ง Rebuild Project รอจนกว่า successfully (หากยังมี error สีแดงในส่วนของการ import อยู่ในบิต แล้วเปิดโปรเจกใหม่อีกรอบ)




- มาที่ Command Prompt แล้วสั่งรันโปรเจกใหม่อีกรอบหนึ่งด้วยคำสั่ง

react-native run-android หรือ **react-native run-ios**

เป็นอันเสร็จเรียบร้อยแล้ว

Administrator: Command Prompt

```
BUILD SUCCESSFUL in 49s
26 actionable tasks: 19 executed, 7 up-to-date
info Running C:\Users\CodingThailand\AppData\Local\And
rm-tools\adb -s emulator-5554 reverse tcp:8081 tcp:808
info Starting the app on emulator-5554 (C:\Users\Codin
ta\Local\Android\Sdk\platform-tools\adb -s emulator-55
rt -n com.myrnproject/com.myrnproject.MainActivity)...
Starting: Intent { cmp=com.myrnproject/.MainActivity }
^CTerminate batch job (Y/N)? y
C:\myrnproject>react-native run-android
```



ทดลองสร้างเพจใหม่ด้วย React Navigation

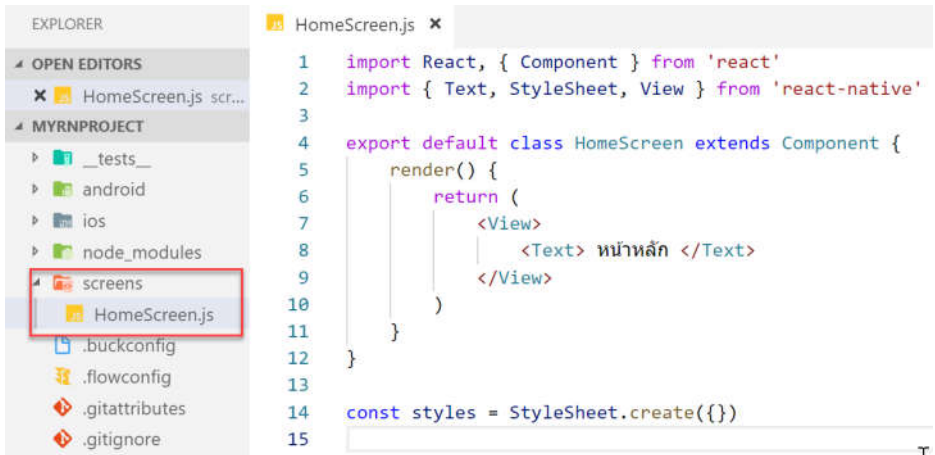
- สร้างโฟลเดอร์ใหม่ตั้งชื่อว่า screens ในโปรเจก ในโฟลเดอร์นี้ให้สร้างไฟล์ชื่อ

HomeScreen.js และเขียนโค้ด ดังนี้

```
import React, { Component } from 'react'
import { Text, StyleSheet, View } from 'react-native'

export default class HomeScreen extends Component {
  render() {
    return (
      <View>
        <Text> หน้าหลัก </Text>
      </View>
    )
  }
}

const styles = StyleSheet.create({})
```



2. แก้ไขไฟล์ **App.js** เพื่อสร้าง Stack Navigation และนำ HomeScreen เข้ามาตั้งค่าและใช้งาน ดังนี้

```

import React, {Component} from 'react';
import {Platform, StyleSheet, Text, View} from 'react-native';
import { createStackNavigator, createAppContainer } from
"react-navigation";

import HomeScreen from './screens/HomeScreen';

const AppNavigator = createStackNavigator({
  Home: HomeScreen
});

const AppContainer = createAppContainer(AppNavigator);

export default class App extends Component {
  render() {
    return <AppContainer />;
  }
}

```

3. เปิดไฟล์ HomeScreen.js อีกครั้งหนึ่ง แก้ไขโค้ดเพื่อทดลองตั้งค่า header bar ให้กับ HomeScreen ดังนี้

```
import React, { Component } from 'react'
import { Text, StyleSheet, View } from 'react-native'

export default class HomeScreen extends Component {
  static navigationOptions = {
    title: 'หน้าหลัก',
    headerStyle: {
      backgroundColor: 'green',
    },
    headerTintColor: '#fff',
    headerTitleStyle: {
      fontWeight: 'bold',
    },
  };

  render() {
    return (
      <View>
        <Text> หน้าหลัก </Text>
      </View>
    )
  }
}

const styles = StyleSheet.create({})
```


4. บันทึกไฟล์ทั้งหมด และกด R สองครั้งต่อเนื่องกันเพื่อดูผลลัพธ์ จะได้ผลลัพธ์ดังรูป

