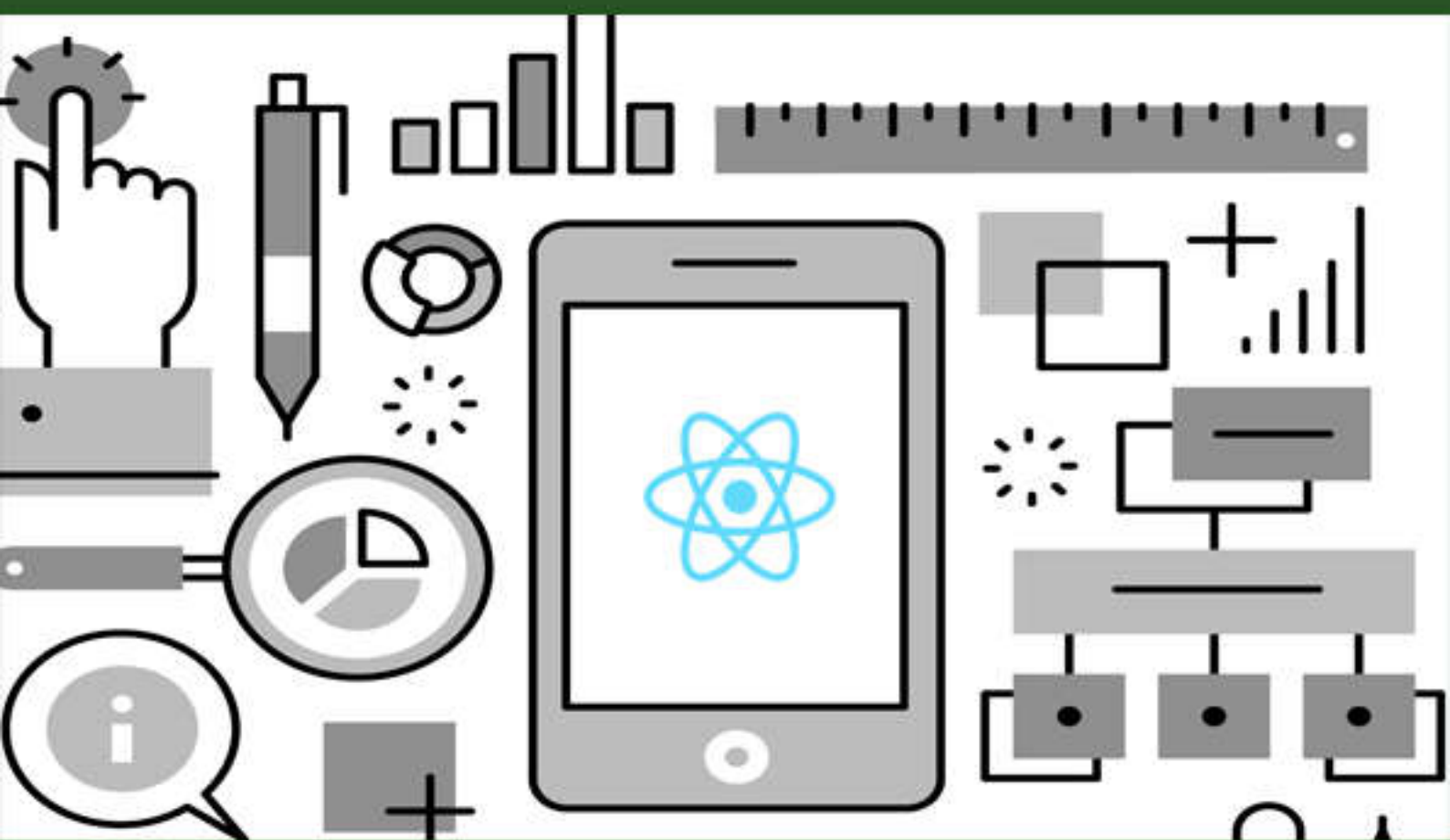


พัฒนา Mobile Application ด้วย **React Native**



โค้ชเอก

CodingThailand

พัฒนา Mobile Application ด้วย React Native

โค้ชเอก

คู่มือเล่มนี้ผู้เขียนตั้งใจจัดทำขึ้นเพื่อใช้ในการอบรมในหลักสูตร “พัฒนา Mobile App ด้วย React Native” ซึ่งเน้นเนื้อหาการเตรียมตัว และติดตั้ง React Native หวังว่าคู่มือเล่มนี้จะเป็นประโยชน์ ประหยัดเวลาการเรียนรู้ของทุกคนครับ

“จงเอาชนะความไม่รู้ ด้วยการพัฒนาตัวเอง และลงมือทำอย่างสม่ำเสมอ”



©2019 โค้ชเอก

สารบัญ

| | |
|---|---------------|
| การเตรียมตัวและการติดตั้ง React Native | 3 |
| - การติดตั้งต้องเตรียมอะไรบ้าง | |
| - การติดตั้ง Android Studio | |
| - ขั้นตอนการสร้าง Emulator | |
| การติดตั้ง React Native | 12 |
| - การติดตั้ง React Native CLI | |
| - การรันโปรเจกต์เพื่อทดสอบ App | |
| - ขั้นตอนการรันโปรเจกต์โดยใช้ Emulator | |
| - ขั้นตอนการรันโปรเจกต์บนเครื่องจริง (USB) | |
| การใช้งาน Visual Studio Code | 16 |
| - การติดตั้ง Extensions สำหรับ React Native | |
| - ทดลองเขียน React Native | |
| สรุปเทคนิคและคำสั่งที่ใช้บ่อย | 20 |
| - การติดตั้ง Extensions สำหรับ React Native | |
| - ทดลองเขียน React Native | |

การเตรียมตัวและการติดตั้ง React Native

การติดตั้งต้องเตรียมอะไรบ้าง

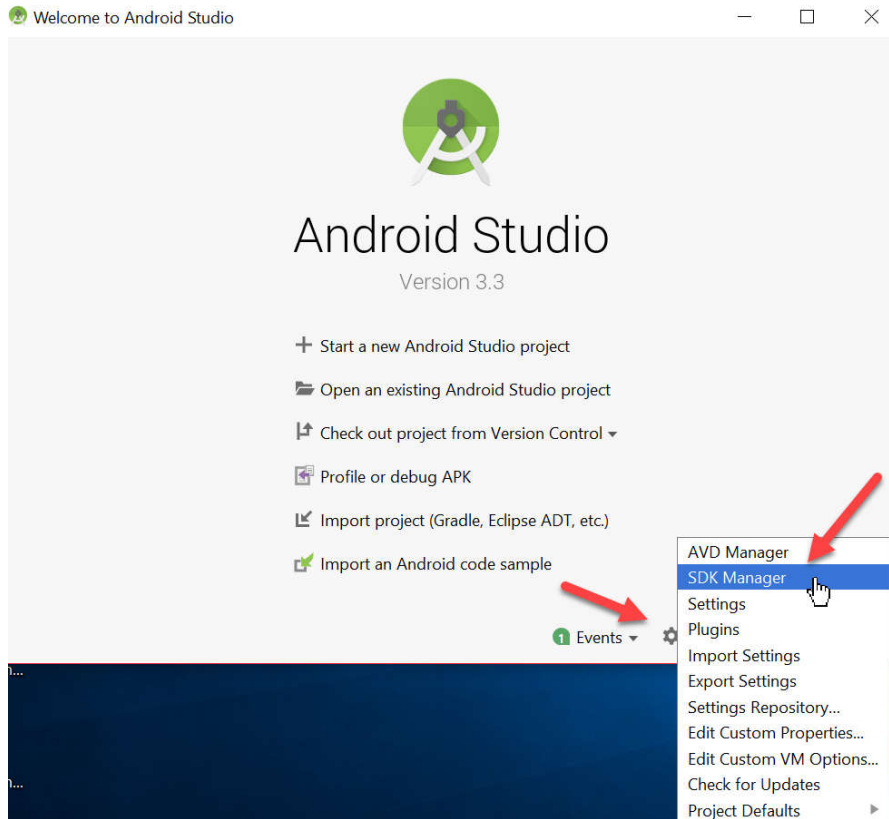
ในการติดตั้ง React Native นั้น ก่อนเรียนต้องเตรียมตัวและติดตั้งโปรแกรมต่างๆ ประกอบด้วย

1. เตรียม Notebook หรือ Mac คนละ 1 เครื่อง สำหรับ Notebook แนะนำเป็น Windows 10 แบบ 64bit ครับ (หรือ อย่างน้อยต้อง Windows 7 64bit) Ram ต้องไม่น้อยกว่า 6 GB (แนะนำ 8 GB ขึ้นไป)
2. ติดตั้ง git ดาวน์โหลดได้ที่นี้ โดยคลิก next อย่างเดียวจนเสร็จเรียบร้อย <https://git-scm.com/download/win>
3. ติดตั้ง Node.js เวอร์ชัน 8 (หรือเวอร์ชัน LTS) โดยคลิก next อย่างเดียวจนเสร็จเรียบร้อย ตามลิงก์ <https://nodejs.org/dist/latest-v8.x/node-v8.15.0-x64.msi>
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 8 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 อีกครั้ง

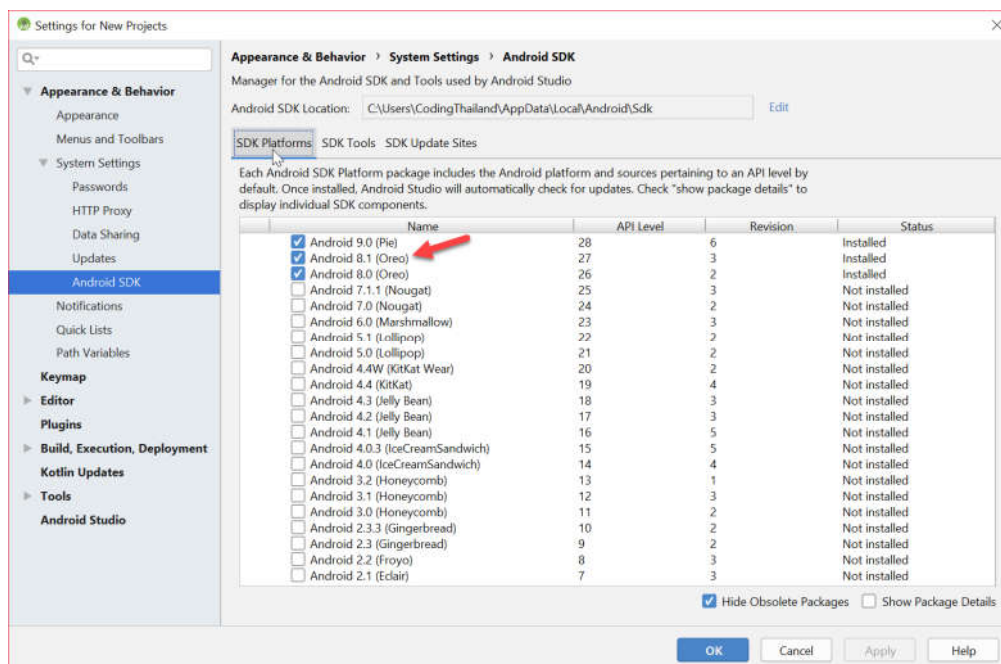
การตั้งค่า Android Studio

หลังจากติดตั้ง Android Studio เรียบร้อย ให้ตั้งค่าและติดตั้ง SDK และ Tools ต่างๆ ดังนี้

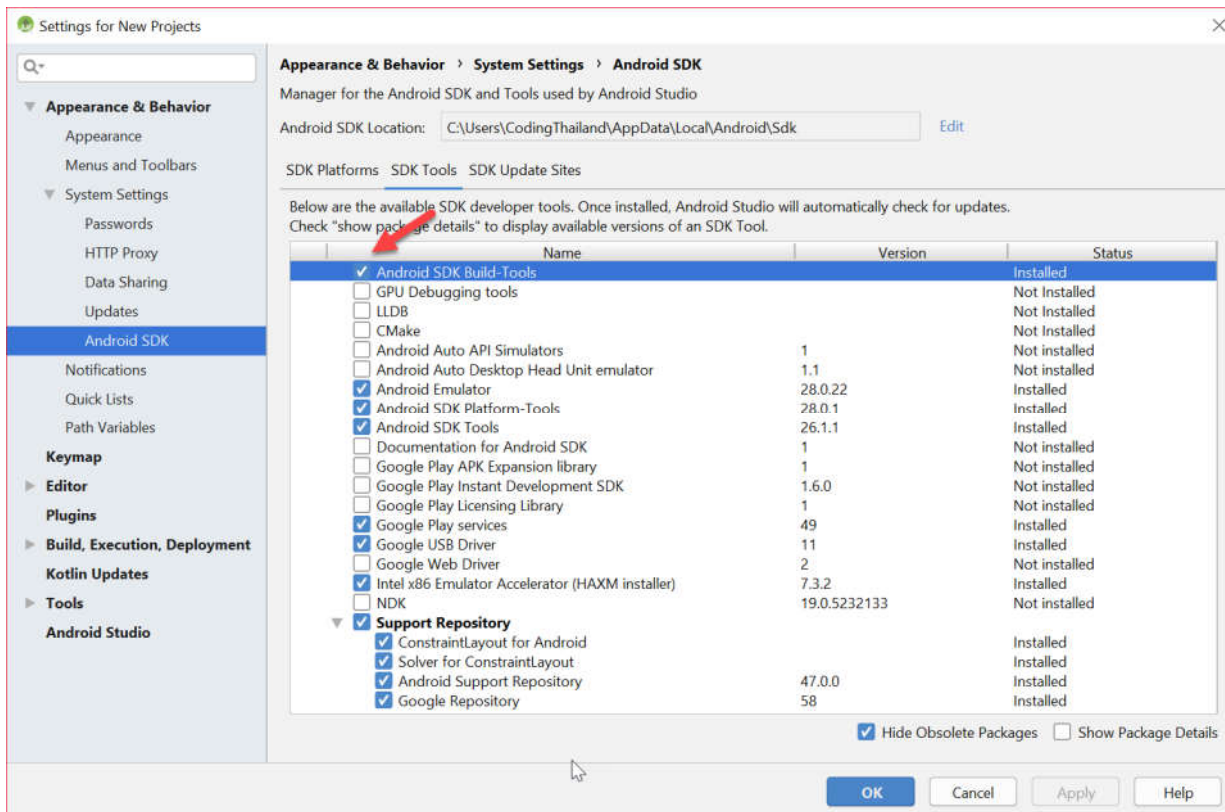
1. เปิด Android Studio ขึ้นมา คลิกเลือกที่เมนู Configure -> SDK Manager



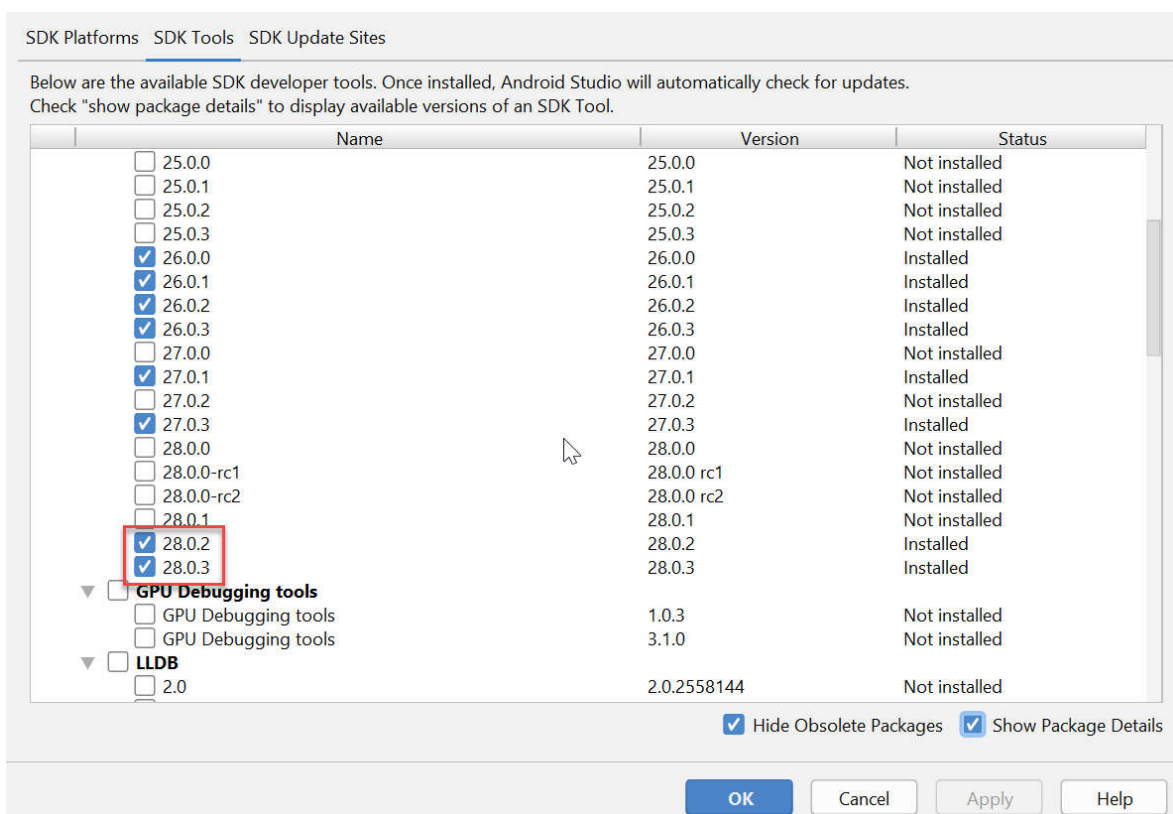
2. ดึงเครื่องหมายถูกที่ Android 8.1 Oreo (API Level 27) หรือใช้ตัวล่าสุดก็ได้เช่นเดียวกัน (Android 9 API Level 28)



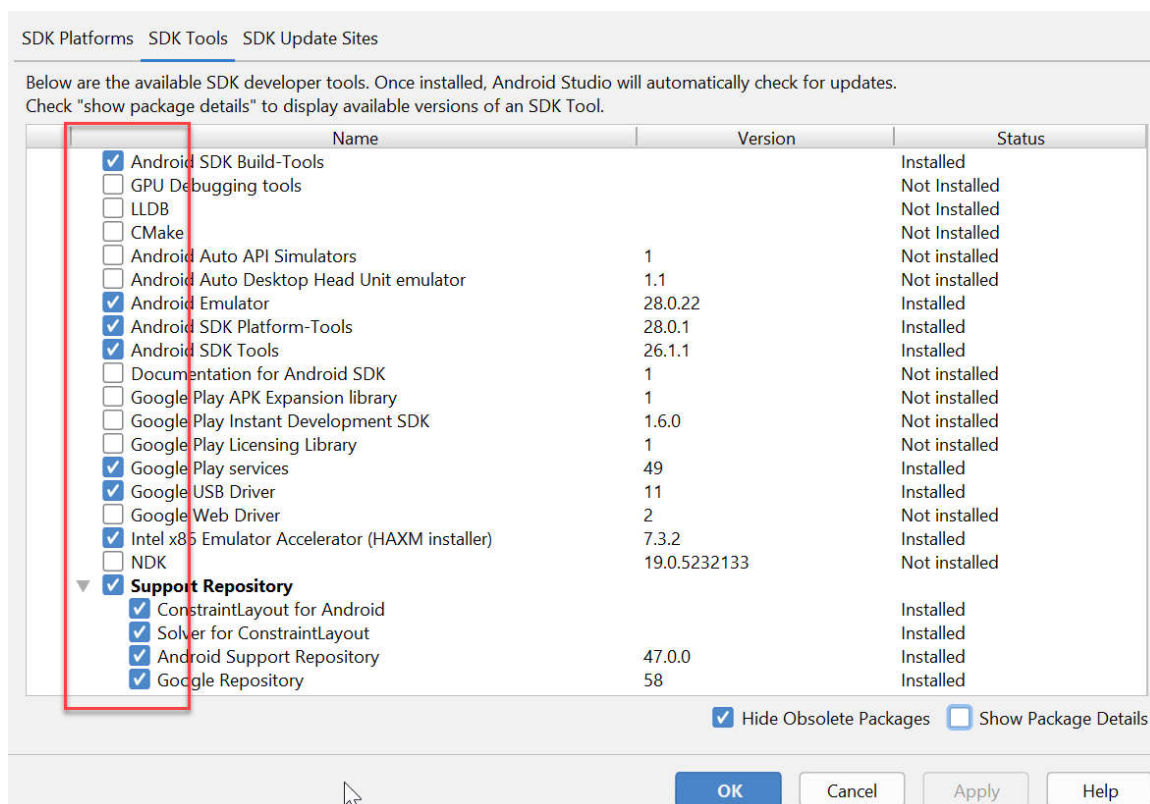
3. คลิกที่ Tab SDK Tools ให้ตรวจสอบว่ามีติดตั้งเครื่องหมายถูกที่ Android SDK Build-Tools เรียบร้อย



4. ให้คลิกเลือกที่ Show Package Details (มุมขวาล่าง) เพื่อตรวจสอบว่าได้ติดตั้งเวอร์ชัน 28.0.2 และ 28.0.3 แล้วหรือไม่ ถ้ายังไม่ได้ติดตั้ง ให้ติ๊ก แล้วคลิกปุ่ม Apply เพื่อทำการติดตั้ง ดังรูป (ขั้นตอนนี้อาจข้ามได้ ปกติตอนรันโปรเจกจะติดตั้งให้อัตโนมัติ)



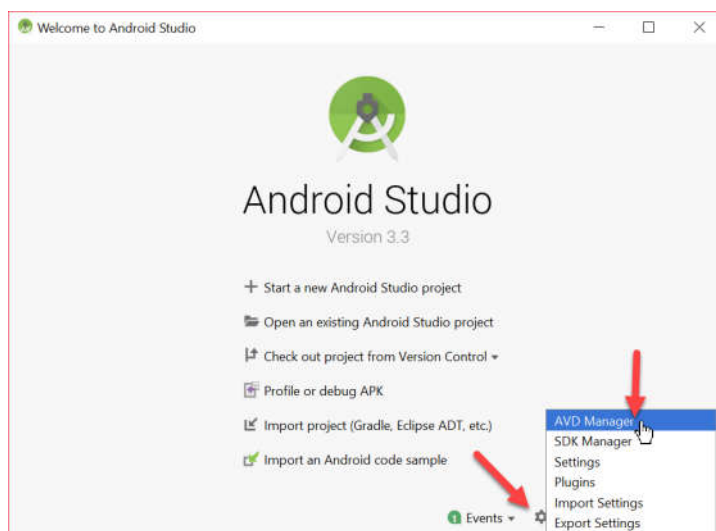
5. สุดท้ายให้ตรวจสอบว่าได้ติดตั้ง หรือติ๊กเครื่องหมายถูกครบตามในภาพนี้แล้วหรือไม่ ถ้ายังไม่ครบ ให้เลือกแล้วกดปุ่ม Apply



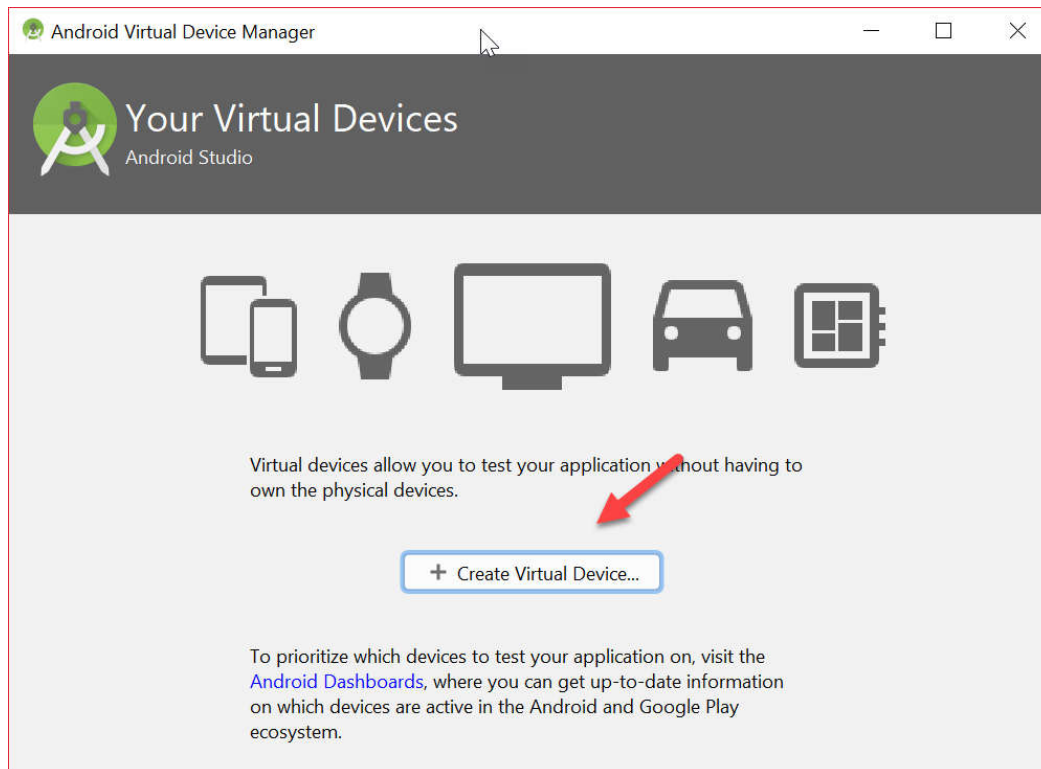
ขั้นตอนการสร้าง Emulator

เราสามารถทดสอบ App ที่เขียนได้โดยใช้เครื่องจริง (usb) หรือจะใช้ emulator ก็ได้เช่นเดียวกันในหัวข้อนี้เราจะมาเรียนรู้ขั้นตอนการสร้าง Emulator กัน

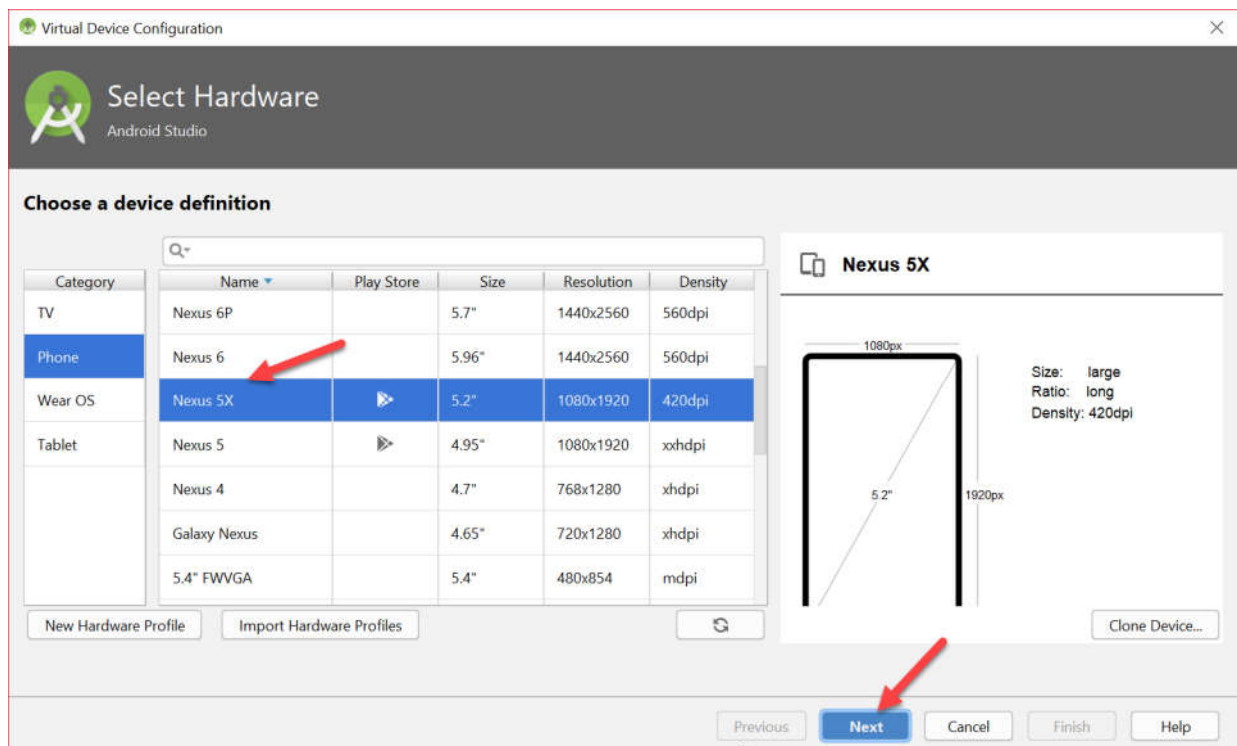
1. เปิด Android Studio ขึ้นมา คลิกเลือกที่เมนู Configure -> AVD Manager



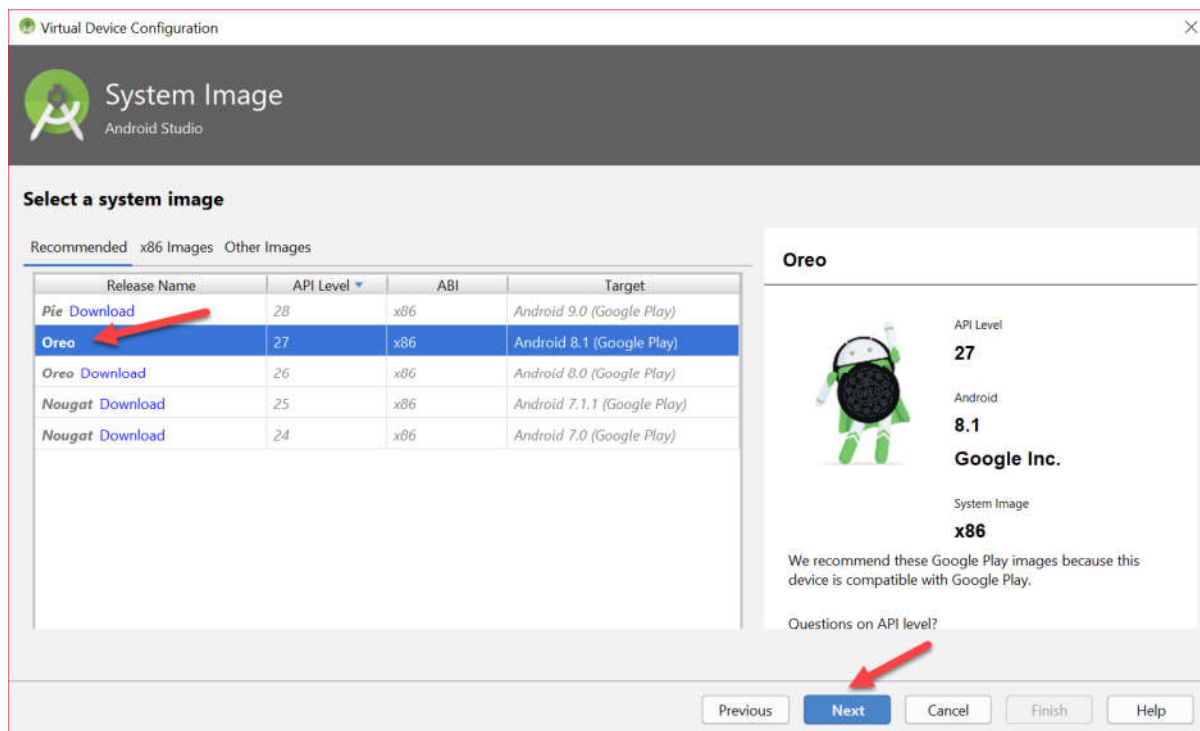
2. คลิกปุ่ม Create Virtual Device...



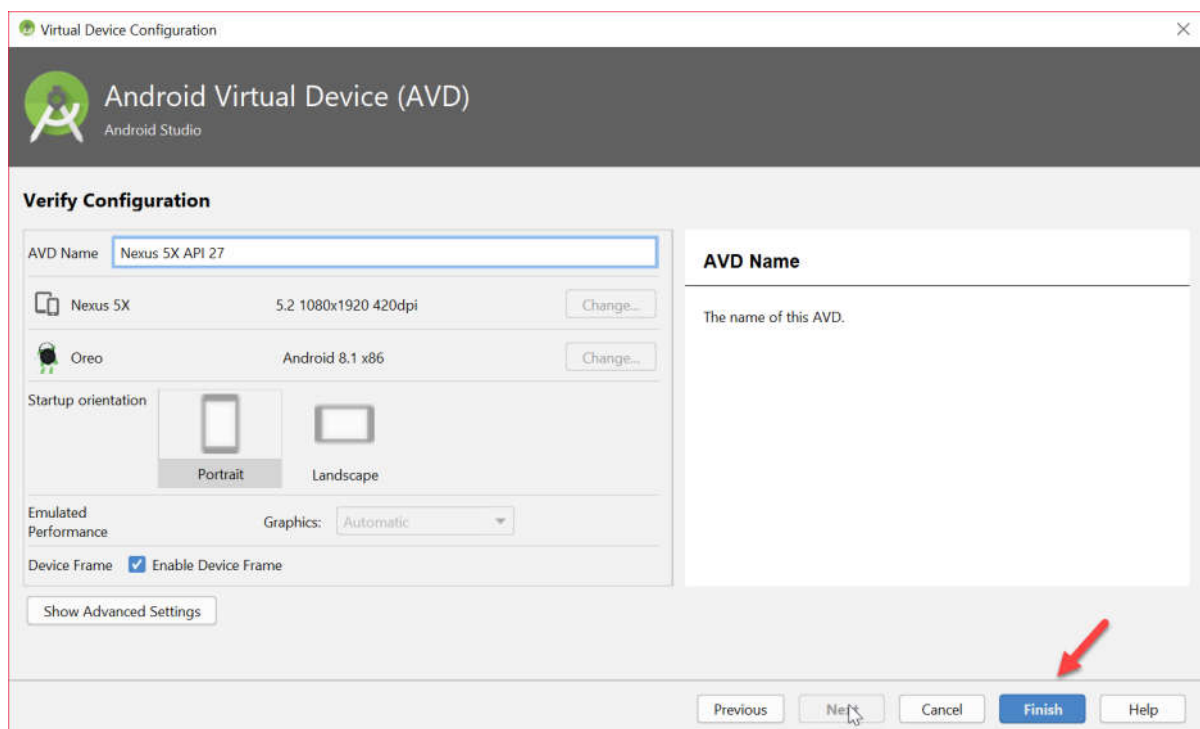
3. คลิกเลือก Nexus 5X แล้วคลิกปุ่ม Next



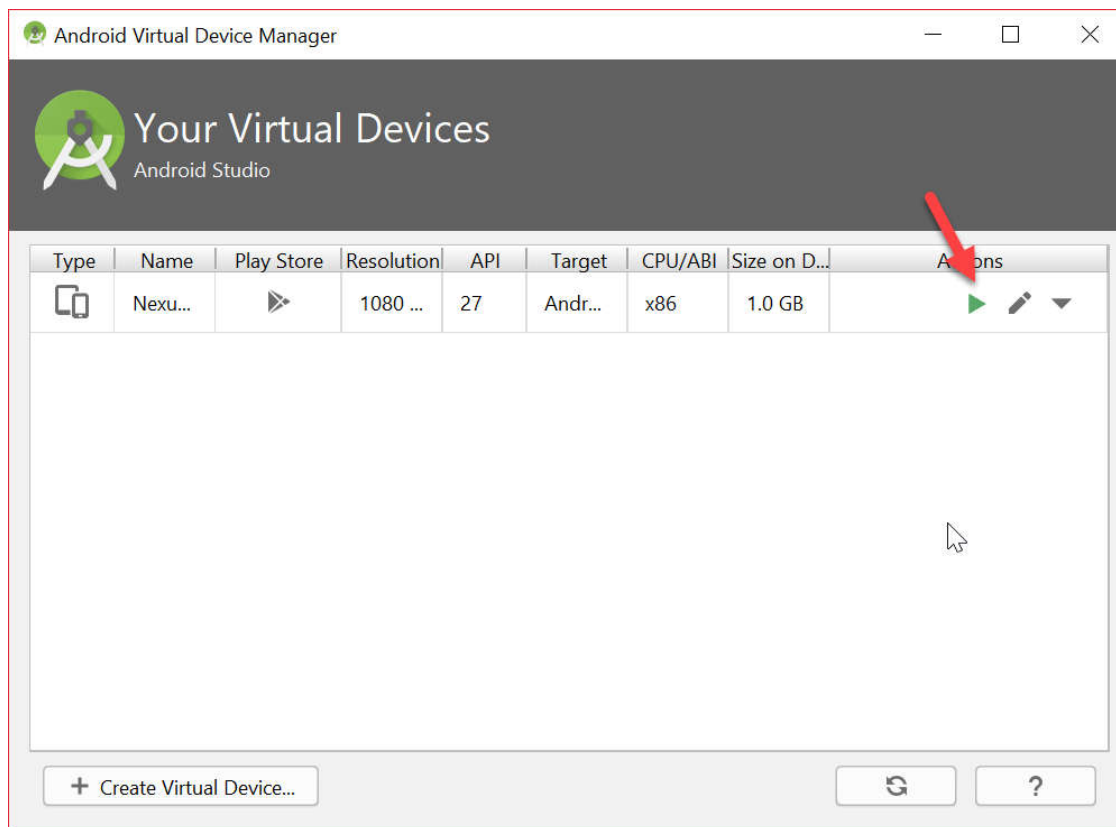
4. คลิกเลือก Oreo (API Level 27) จากนั้นคลิกปุ่ม Next (จริงๆ หากอยากเลือกล่าสุดก็ได้เช่นเดียวกัน)



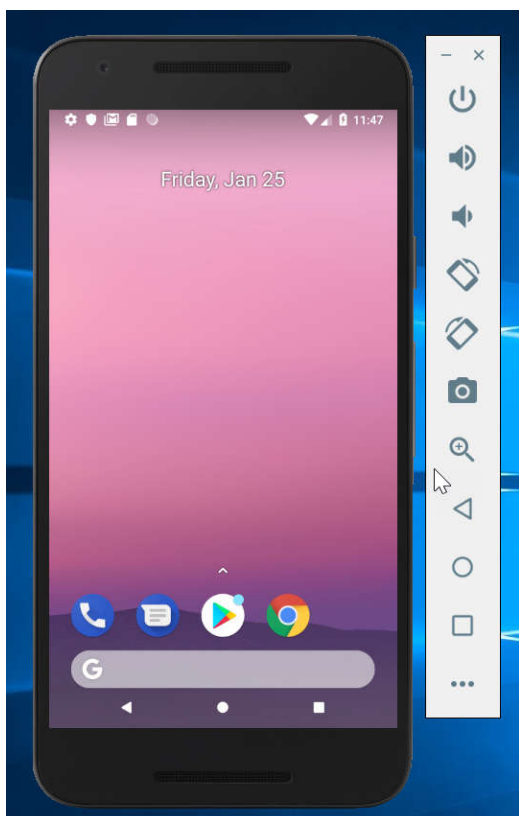
5. คลิกปุ่ม Finish



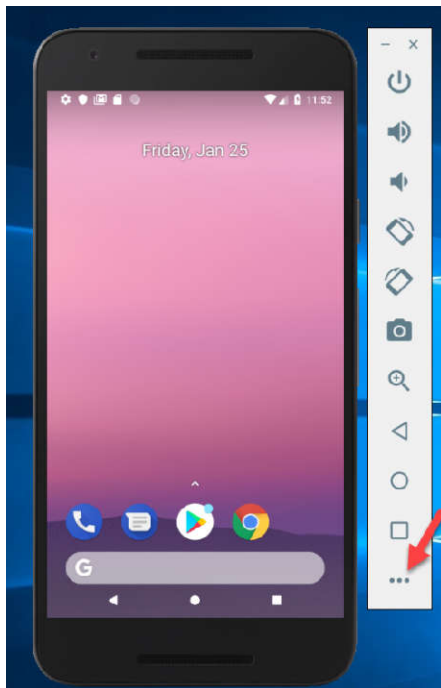
6. คลิกปุ่มสีเขียว เพื่อรัน Emulator ดังรูป



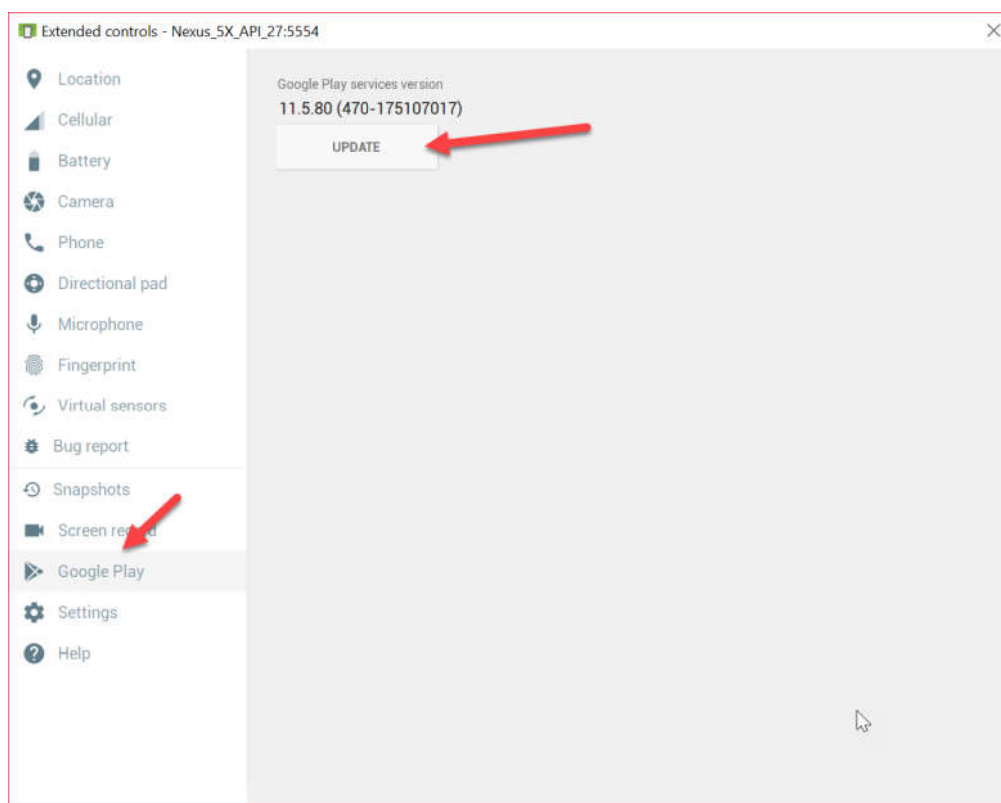
7. เมื่อกดรันแล้วจะได้ Emulator ดังรูป เป็นอันเสร็จเรียบร้อย



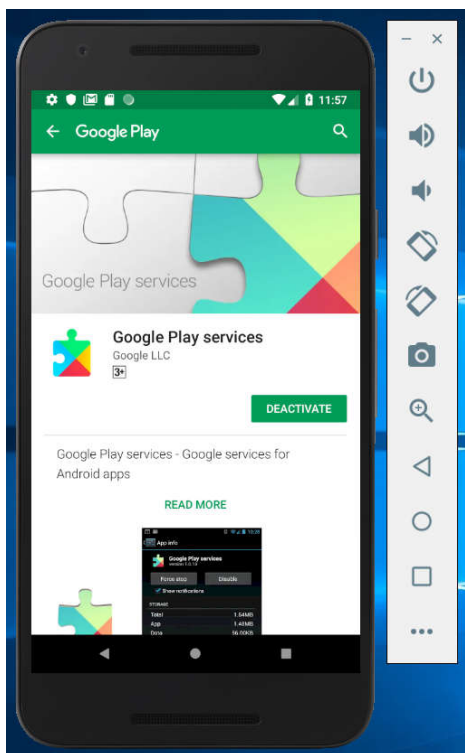
8. หลังจากได้ Emulator มาแล้วควรอัปเดต Google Play Services ให้ล่าสุดด้วย เพื่อในอนาคตเราต้องการใช้ Google Maps API, Location APIs เป็นต้น ให้คลิกที่เมนู ดังรูป



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



10. ให้ล็อกอินเข้าระบบของ Google Play รวบรวมข้อมูล Google Play Services จะได้รับการอัปเดตเป็นเวอร์ชันล่าสุดเรียบร้อยแล้ว



Note: เพื่อความแน่ใจสามารถไปเช็คเลขเวอร์ชันได้อีกครั้งว่าได้รับการอัปเดตหรือไม่


การติดตั้ง React Native

การติดตั้ง React Native CLI

หลังจากที่ได้ติดตั้งโปรแกรมต่างๆ ครบแล้ว ต่อไปจะเป็นขั้นตอนการติดตั้ง React Native CLI

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 ให้คลิกขวาแล้วเลือก Run as administrator ด้วย

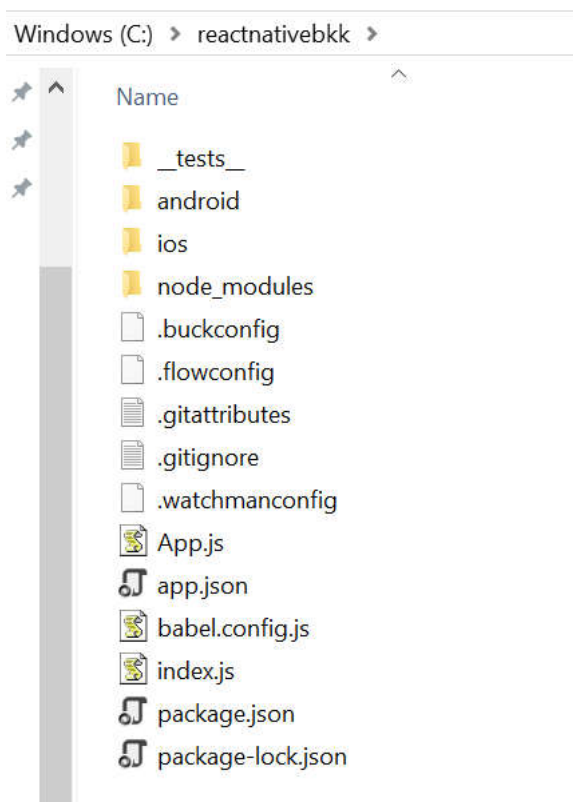
Note: สำหรับเครื่อง Mac ต้องใส่ sudo ข้างหน้าคำสั่งด้วย

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

 npm

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

- เมื่อติดตั้งเสร็จแล้วจะได้โฟลเดอร์ใหม่เป็นชื่อโปรเจกของเรา และไฟล์โค้ดต่างๆ จะถูกเก็บอยู่ในโฟลเดอร์นี้ ดังรูป

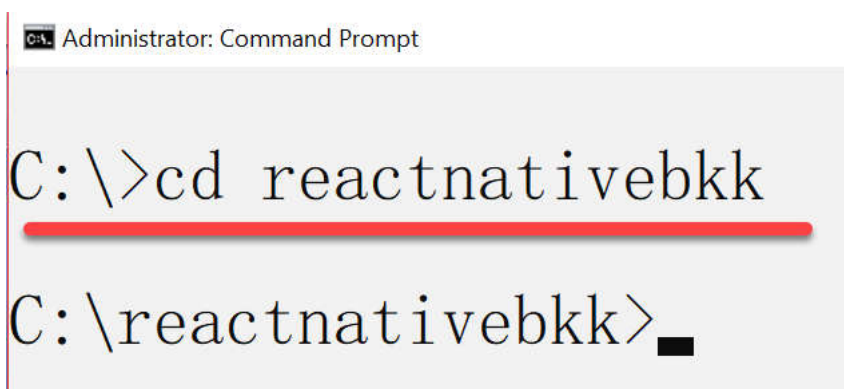


การรันโปรเจกเพื่อทดสอบ App

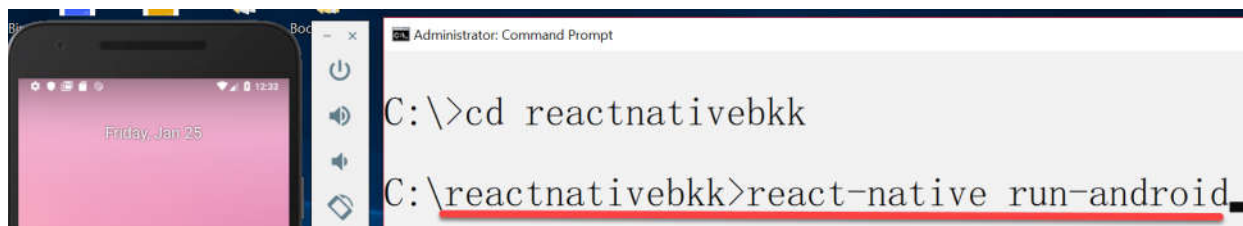
การทดสอบ App นั้นทำได้ 2 วิธีหลักๆ ด้วยกัน ได้แก่ การรันผ่าน Emulator และ การรันโดยใช้เครื่องจริง (usb)

ขั้นตอนการรันโปรเจกโดยใช้ Emulator

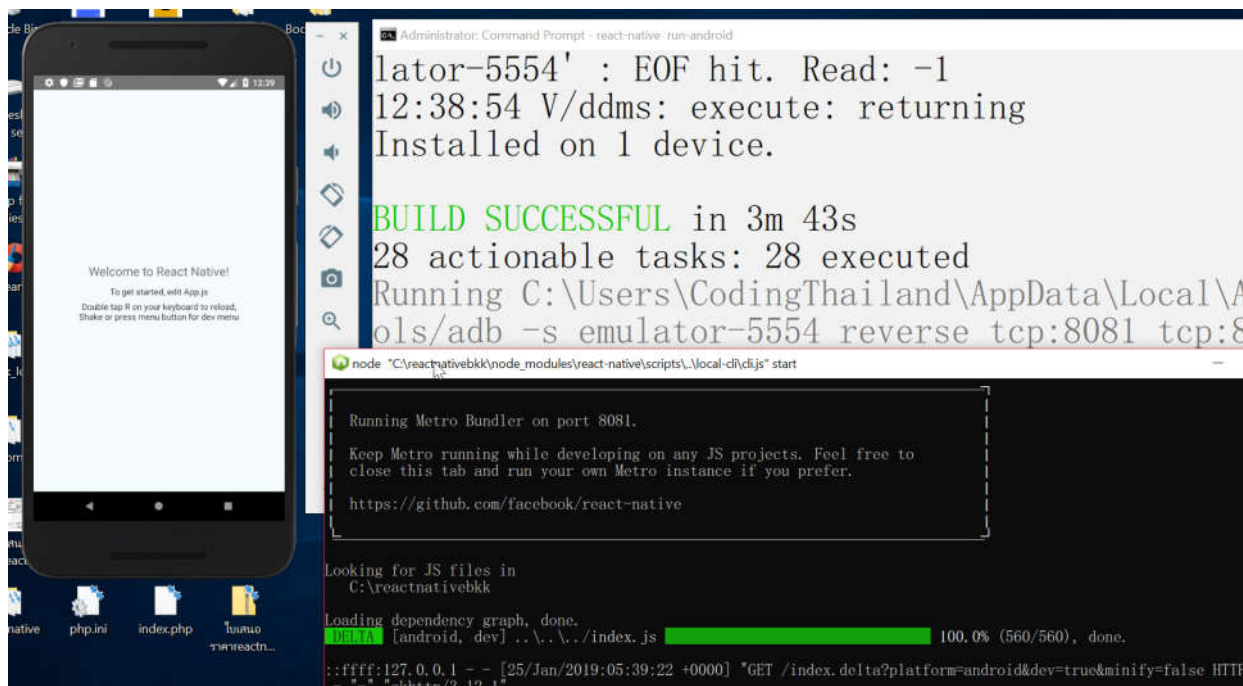
- เปิด Command Prompt ใช้คำสั่ง `cd` เพื่อเข้าไปในโฟลเดอร์โปรเจก ดังรูป



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

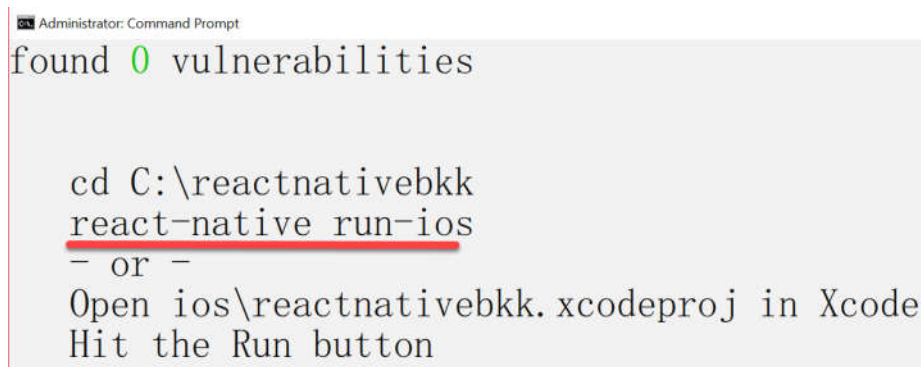


3. รอสักครู่ โปรเจกต์จะถูกนำไปรันใน Emulator เรียบร้อย ดังรูป



Note: หากรันขั้นตอนที่ไม่ผ่าน แนะนำให้เปิดโปรเจกต์ (โฟลเดอร์ android) ใน Android Studio เพื่อ build ก่อน ค่อยกลับมารันอีกครั้งหนึ่ง

Note: สำหรับเครื่อง MAC ให้ใช้คำสั่ง `react-native run-ios` หรือเปิดโปรแกรม Xcode แล้วเปิดไฟล์ในโฟลเดอร์ `ios\reactnativebkk.xcodeproj` จากนั้นก็กดปุ่มรันตามปกติ



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. เมื่อเสียบสายโทรศัพท์เรียบร้อยแล้ว สามารถตรวจสอบว่าโทรศัพท์ได้เชื่อมต่อแล้วหรือไม่ ด้วยคำสั่ง adb devices หรือจะเปิด Android Studio แล้วกดรันตรวจการเชื่อมต่อก็ได้เช่นเดียวกัน (จะเห็นชื่ออุปกรณ์แสดงขึ้นมา)
3. ใช้คำสั่ง react-native run-android เพื่อรันตามปกติ

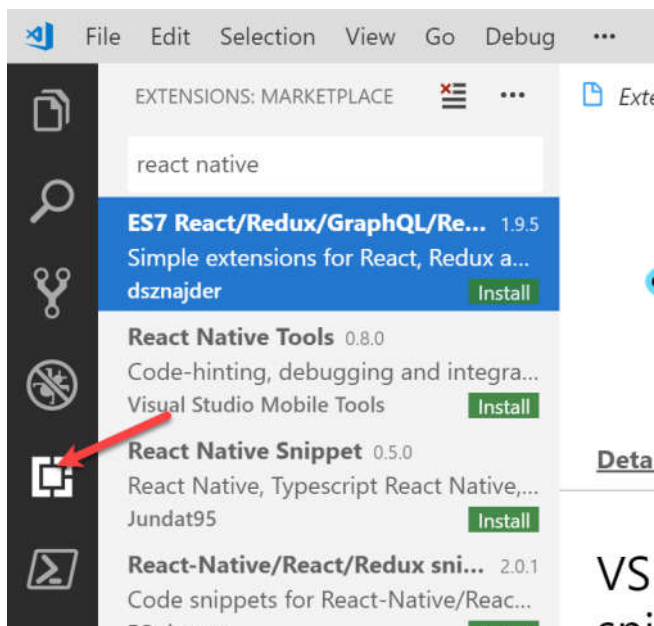
Note: สำหรับ Mac ดูขั้นตอนเพิ่มเติมได้ที่ <https://facebook.github.io/react-native/docs/running-on-device>

การใช้งาน Visual Studio Code

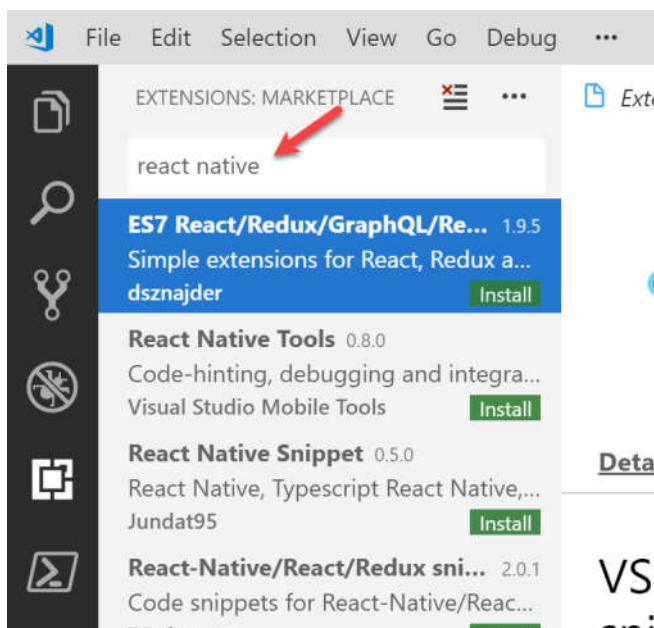
ในหัวข้อนี้เราจะมีติดตั้ง Extensions เพื่อช่วยให้การเขียน React Native ได้ง่ายขึ้น

การติดตั้ง Extensions สำหรับ React Native

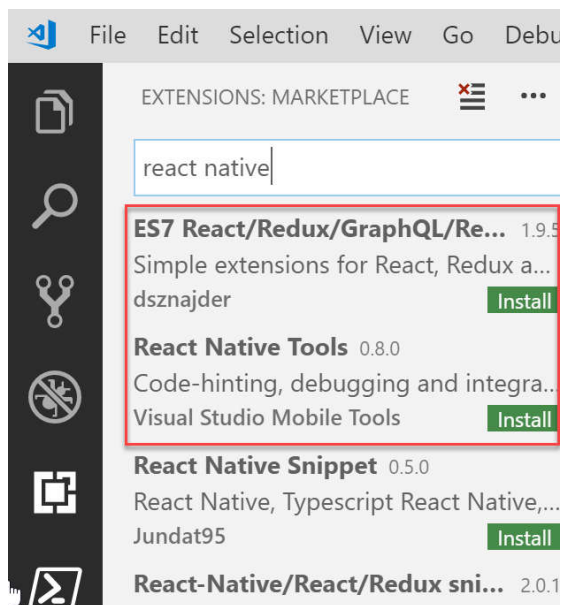
1. เปิดโปรแกรม Visual Studio Code คลิกที่ไอคอน Extensions เพื่อติดตั้ง React Native Extensions



2. พิมพ์คำว่า react native เพื่อค้นหา ดังรูป



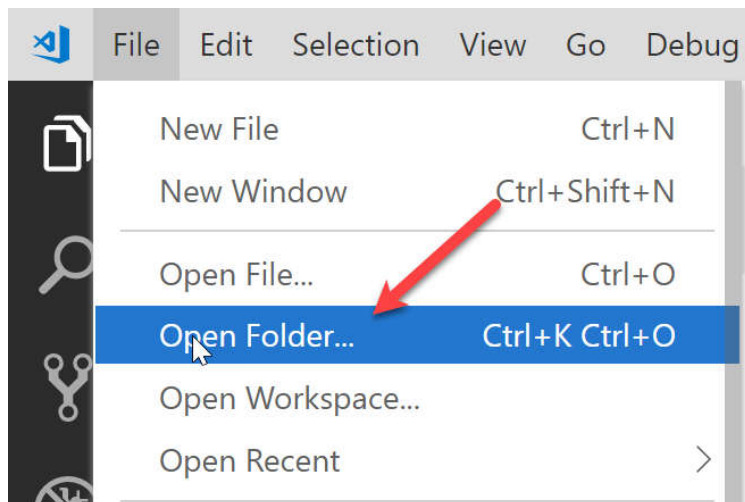
3. คลิกปุ่ม Install (สีเขียว) 2 extensions ได้แก่ ES7 React/Redux.. และ React Native Tools ดังรูป



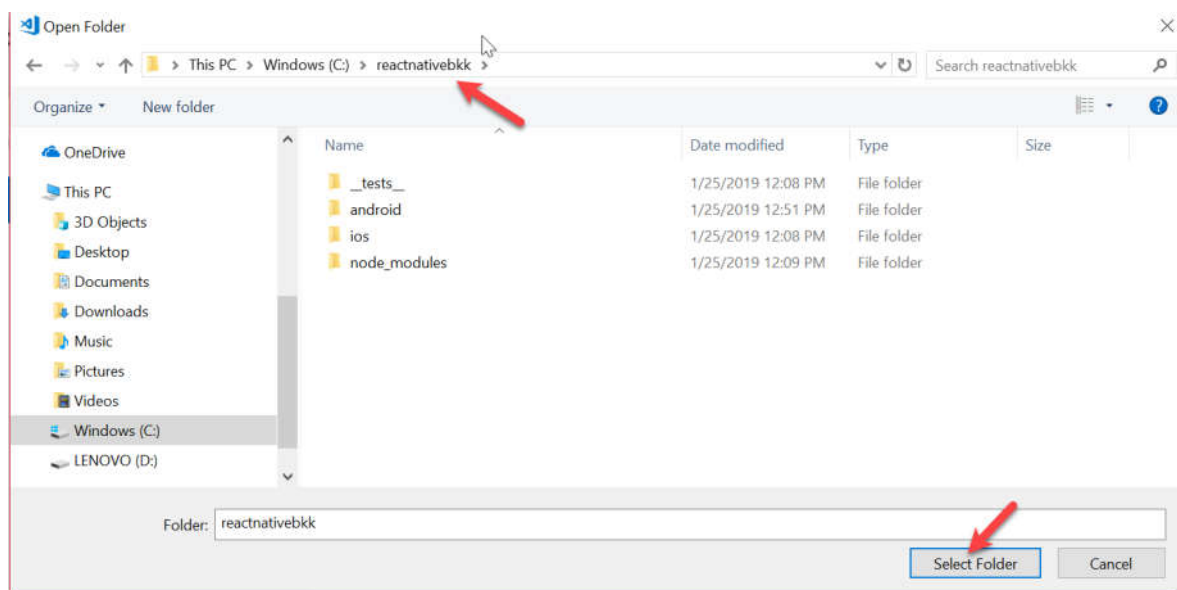
ทดลองเขียน React Native

ในหัวข้อนี้เราจะมาลองแก้ไขโค้ด เพื่อแสดง Hello World กัน

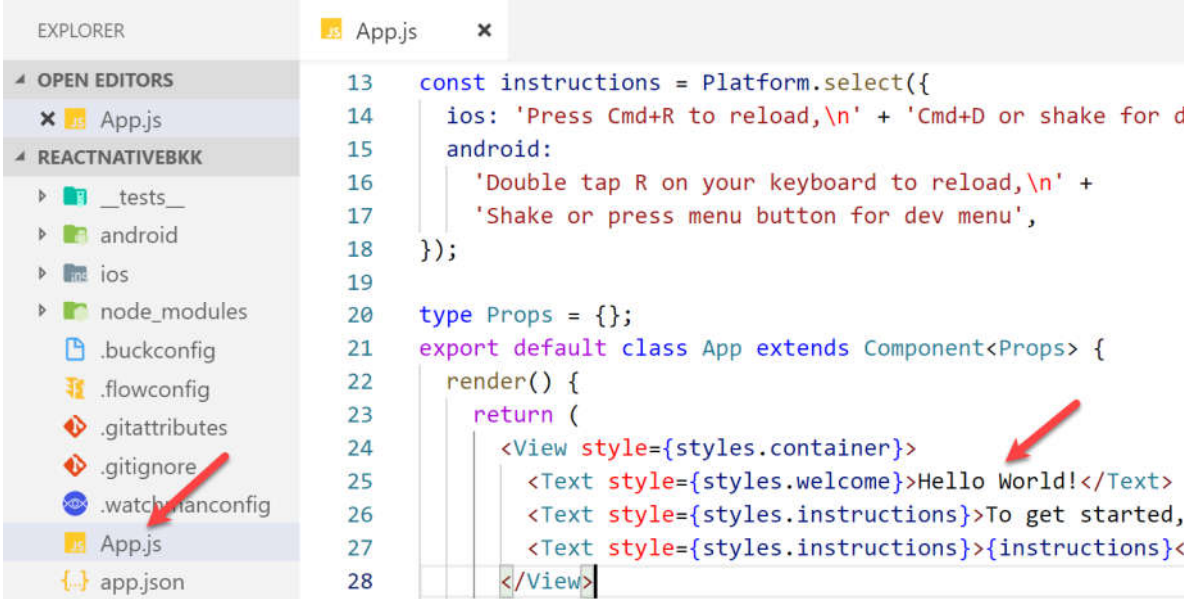
1. ใช้ Visual Studio Code เพื่อเปิดโฟลเดอร์โปรเจกต์ คลิกที่เมนู File -> Open Folder...



2. คลิกเลือกโฟลเดอร์โปรเจกต์



3. เปิดไฟล์ App.js แล้วเขียนโค้ด Hello World ดังรูป แล้วบันทึกไฟล์

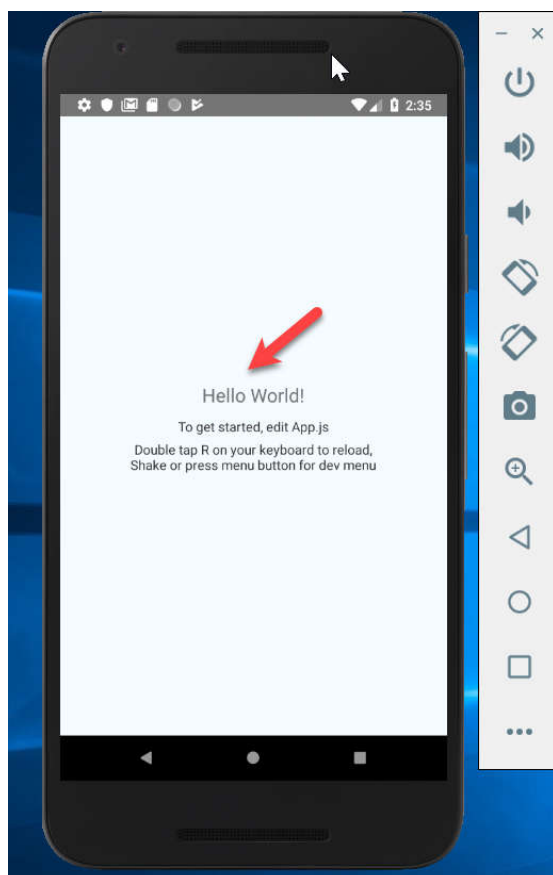


```

13  const instructions = Platform.select({
14    ios: 'Press Cmd+R to reload,\n' + 'Cmd+D or shake for d
15    android:
16      'Double tap R on your keyboard to reload,\n' +
17      'Shake or press menu button for dev menu',
18  });
19
20  type Props = {};
21  export default class App extends Component<Props> {
22    render() {
23      return (
24        <View style={styles.container}>
25          <Text style={styles.welcome}>Hello World!</Text>
26          <Text style={styles.instructions}>To get started,
27          <Text style={styles.instructions}>{instructions}<
28        </View>

```

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



สรุปเทคนิคและคำสั่งที่ใช้บ่อย

1. การสร้างโปรเจกต์ใหม่

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

2. การรันโปรเจกต์

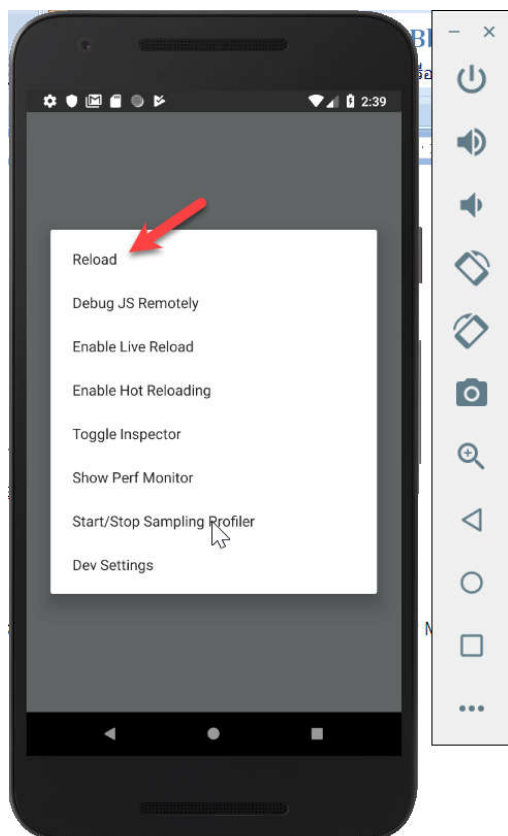
สำหรับ android ใช้คำสั่ง `react-native run-android`

สำหรับ ios ใช้คำสั่ง `react-native run-ios`

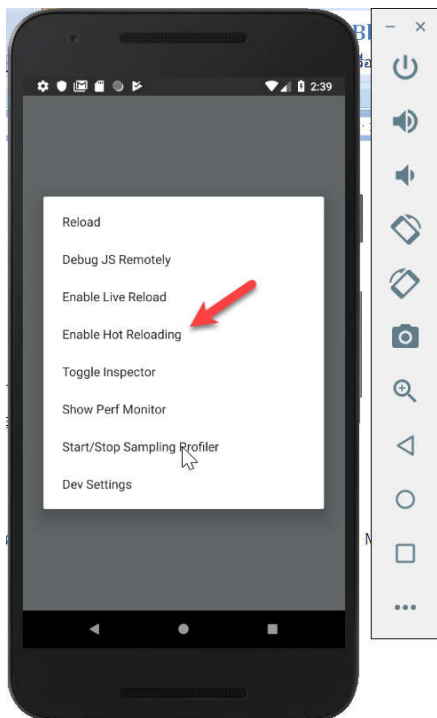
3. เมื่อแก้ไขโค้ดแล้วอยากดูผลลัพธ์

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

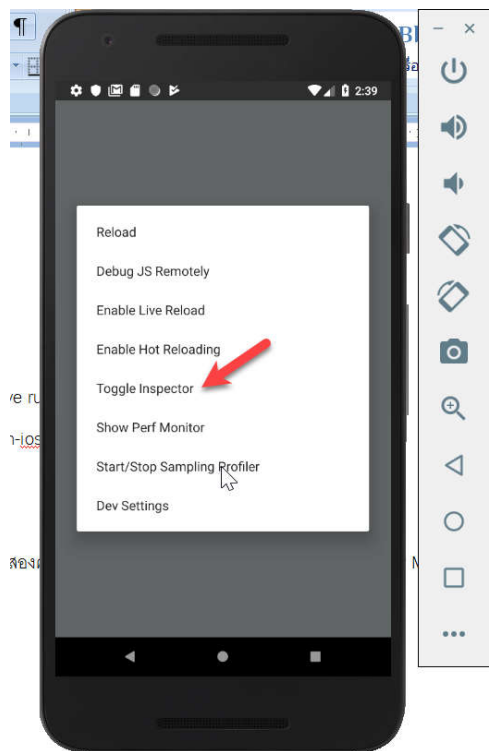
สำหรับ ios ให้กด `Cmd+R` เพื่อ reload หรือกด `Cmd+D` เพื่อเปิด Developer Menu



4. หากต้องการใช้งาน Hot Reloading (เมื่อ save ไฟล์โค้ดที่เราแก้ ตัว App จะเปลี่ยนและแสดงผล ณ จุดนั้นๆทันที โดยไม่ต้องโหลดใหม่ทั้งหมด) ให้ Ctrl + M หรือ Cmd+D เพื่อเปิด Developer Menu จากนั้นคลิกเลือกที่ Enable Hot Reloading ดังรูป

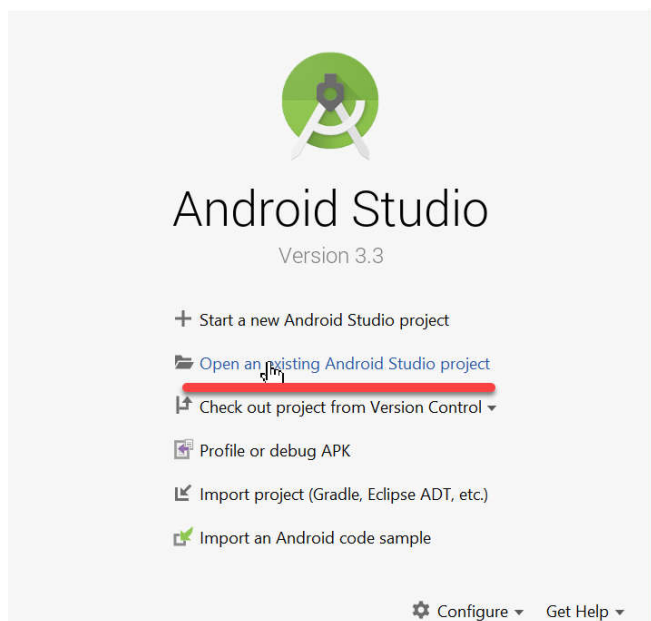


5. การดูค่า CSS ที่เขียนหรือเขียนเพื่อออกแบบหน้าจอ ให้เลือก Toggle Inspector ดังรูป

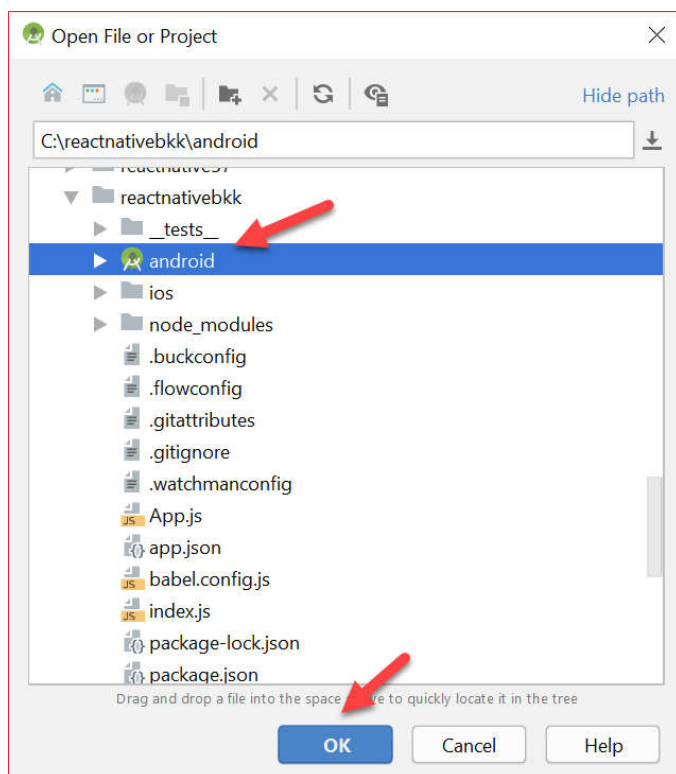


6. การเปิดโปรเจกใน Android Studio

5.1 ให้เปิดโปรแกรม Android Studio ขึ้นมา คลิกเลือกที่ Open and existing Android Studio project



6.2 เลือกโฟลเดอร์โปรเจก แล้วเลือกที่ android จากนั้นคลิกปุ่ม OK ก็เป็นอันเสร็จเรียบร้อย



เมื่อเปิดโปรเจกใน Android Studio เราจะสามารถ build หรือสร้างไฟล์ apk จากที่นี่ได้เลย

Note: สำหรับ Mac ให้ใช้โปรแกรม xCode โดยเปิดโปรเจกในโฟลเดอร์ ios แทน