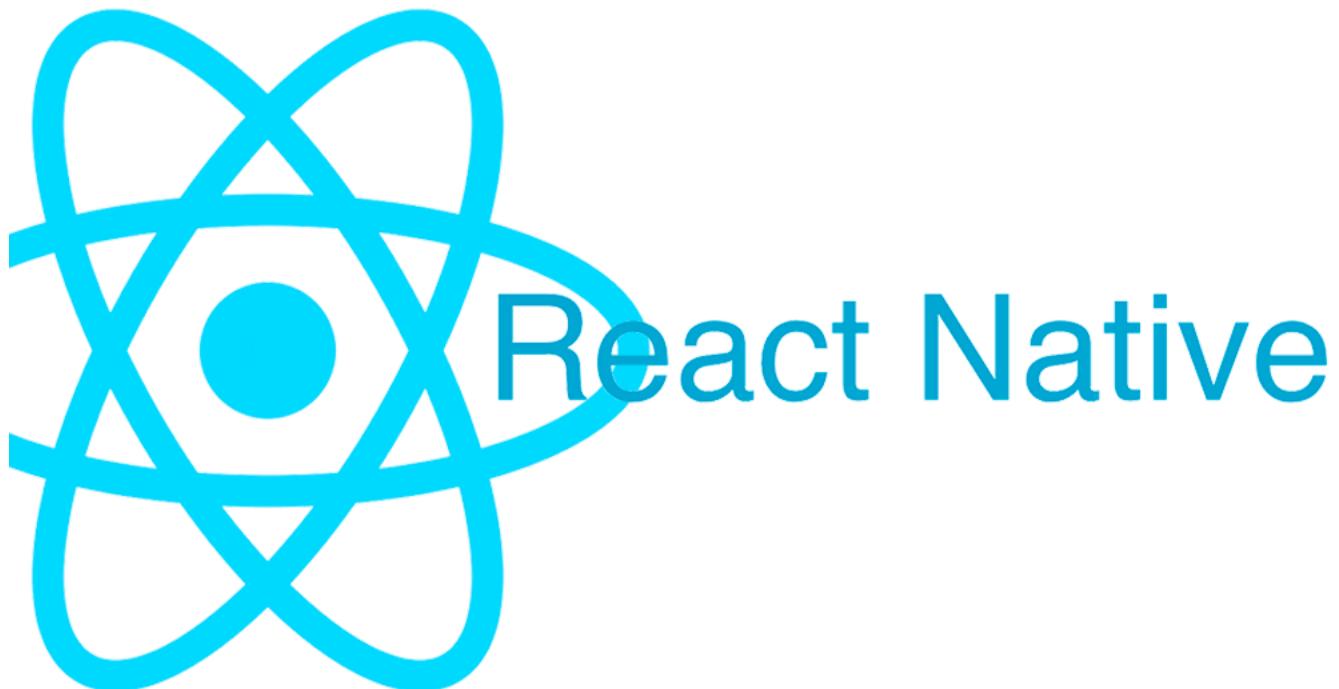


# React Native Setup

Software Studio

DataLab, CS, NTHU

# React Native



Flutter



Flutter

# Setting up the development environment

- React Native CLI
- Expo CLI

# React Native CLI - Installation

- Requirements
  - JDK 8
  - Python: both 2, 3 works for react native
    - Window:
      - Install Python2 and JDK through [Chocolatey](#)
      - `choco install -y nodejs.install python2 jdk8`
    - Mac:
      - Using Homebrew
        - `brew cask install adoptopenjdk/openjdk/adoptopenjdk8`
      - Modern macOS versions come with Python 2.7.x itself

[\[source\]](#)

# React Native CLI - Android development environment

- Android Studio

- Choose a "Custom" setup
- Make sure the boxes are checked:
  - Android SDK
  - Android SDK Platform
  - Performance(Intel HAXM)
  - Android Virtual Device

# React Native CLI - Android development environment

- Install the Android SDK
  - open SDK Manager
  - Windows:
    - File->settings->Appearance & Behavior->Android SDK
  - Mac:
    - Preferences->Appearance & Behavior → System Settings → Android SDK
  - make sure the following items are checked
    - Android SDK Platform 28
    - Intel x86 Atom\_64 System Image or Google APIs Intel x86 Atom System Image •
  - then click Apply

BlurlImage [C:\Users\alan\AndroidStudioProjects\BlurlImage] - BlurlImage - Android Studio

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window

New Open... Profile or Debug APK... Open Recent Close Project Settings... Ctrl+Alt+S Project Structure... Ctrl+Alt+Shift+S Other Settings Import Settings... Export Settings... Settings Repository... Export to Zip File... Sync Project with Gradle Files Re-Import Gradle Project Save All Ctrl+S Sync with File System Ctrl+Alt+Y Invalidate Caches / Restart... Export to HTML... Print... Add to Favorites File Encoding Remove BOM

Appearance & Behavior

Appearance Menus and Toolbars

System Settings

Passwords HTTP Proxy Data Sharing Updates

Android SDK

File Colors Scopes Notifications Quick Lists Path Variables

Keymap

Editor Plugins

Version Control

Build, Execution, Deployment

Languages & Frameworks

Tools

Other Settings

Experimental

Settings

Appearance & Behavior > System Settings > Android SDK

Manager for the Android SDK and Tools used by Android Studio

Android SDK Location: C:\Users\alan\AppData\Local\Android\Sdk Edit

SDK Platforms SDK Tools SDK Update Sites

Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components.

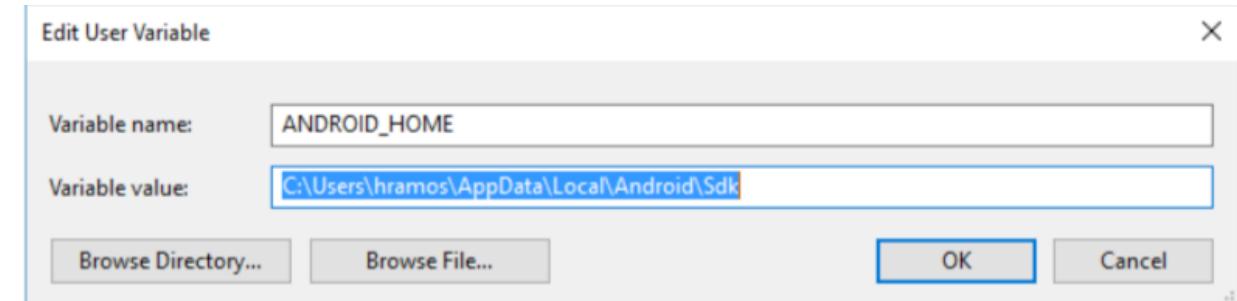
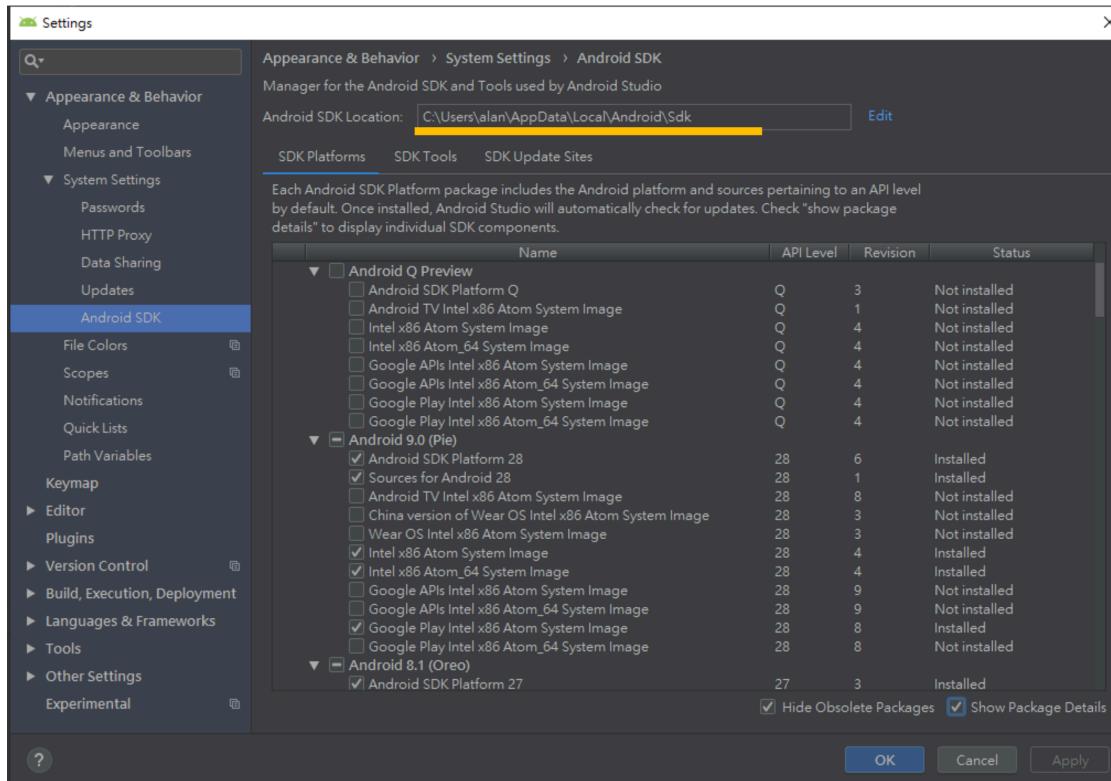
	Name	API Level	Revision	Status
Android Q Preview				
Android SDK Platform Q	Q	3		Not installed
Android TV Intel x86 Atom System Image	Q	1		Not installed
Intel x86 Atom System Image	Q	4		Not installed
Intel x86 Atom_64 System Image	Q	4		Not installed
Google APIs Intel x86 Atom System Image	Q	4		Not installed
Google APIs Intel x86 Atom_64 System Image	Q	4		Not installed
Google Play Intel x86 Atom System Image	Q	4		Not installed
Google Play Intel x86 Atom_64 System Image	Q	4		Not installed
Android 9.0 (Pie)				
Android SDK Platform 28	28	6		Installed
Sources for Android 28	28	1		Installed
Android TV Intel x86 Atom System Image	28	8		Not installed
China version of Wear OS Intel x86 Atom System Image	28	3		Not installed
Wear OS Intel x86 Atom System Image	28	3		Not installed
Intel x86 Atom System Image	28	4		Installed
Intel x86 Atom_64 System Image	28	4		Installed
Google APIs Intel x86 Atom System Image	28	9		Not installed
Google APIs Intel x86 Atom_64 System Image	28	9		Not installed
Google Play Intel x86 Atom System Image	28	8		Installed
Google Play Intel x86 Atom_64 System Image	28	8		Not installed
Android 8.1 (Oreo)				
Android SDK Platform 27	27	3		Installed

Hide Obsolete Packages  Show Package Details

OK Cancel Apply

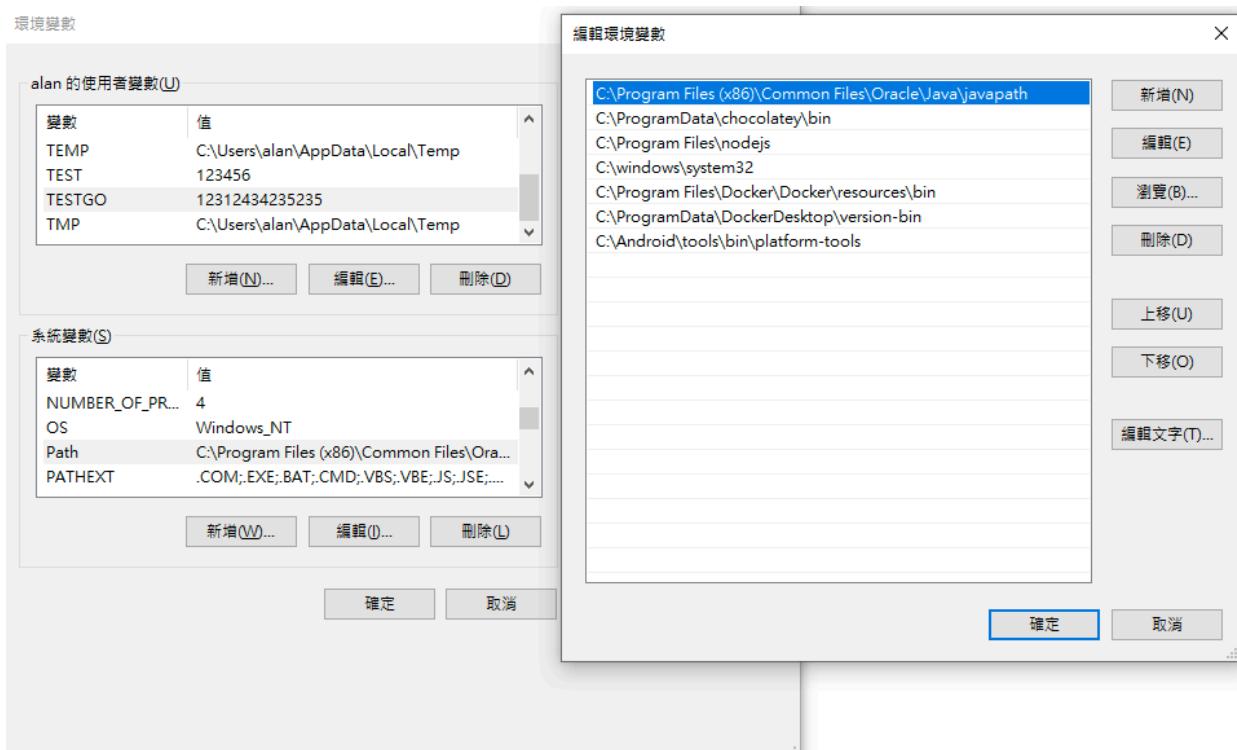
# React Native CLI - Android development environment

- Windows:
  - Configure the ANDROID\_HOME environment variable



# React Native CLI - Android development environment

- Windows:
- Add platform-tools to Path
  - C:\Android\tools\bin\platform-tools

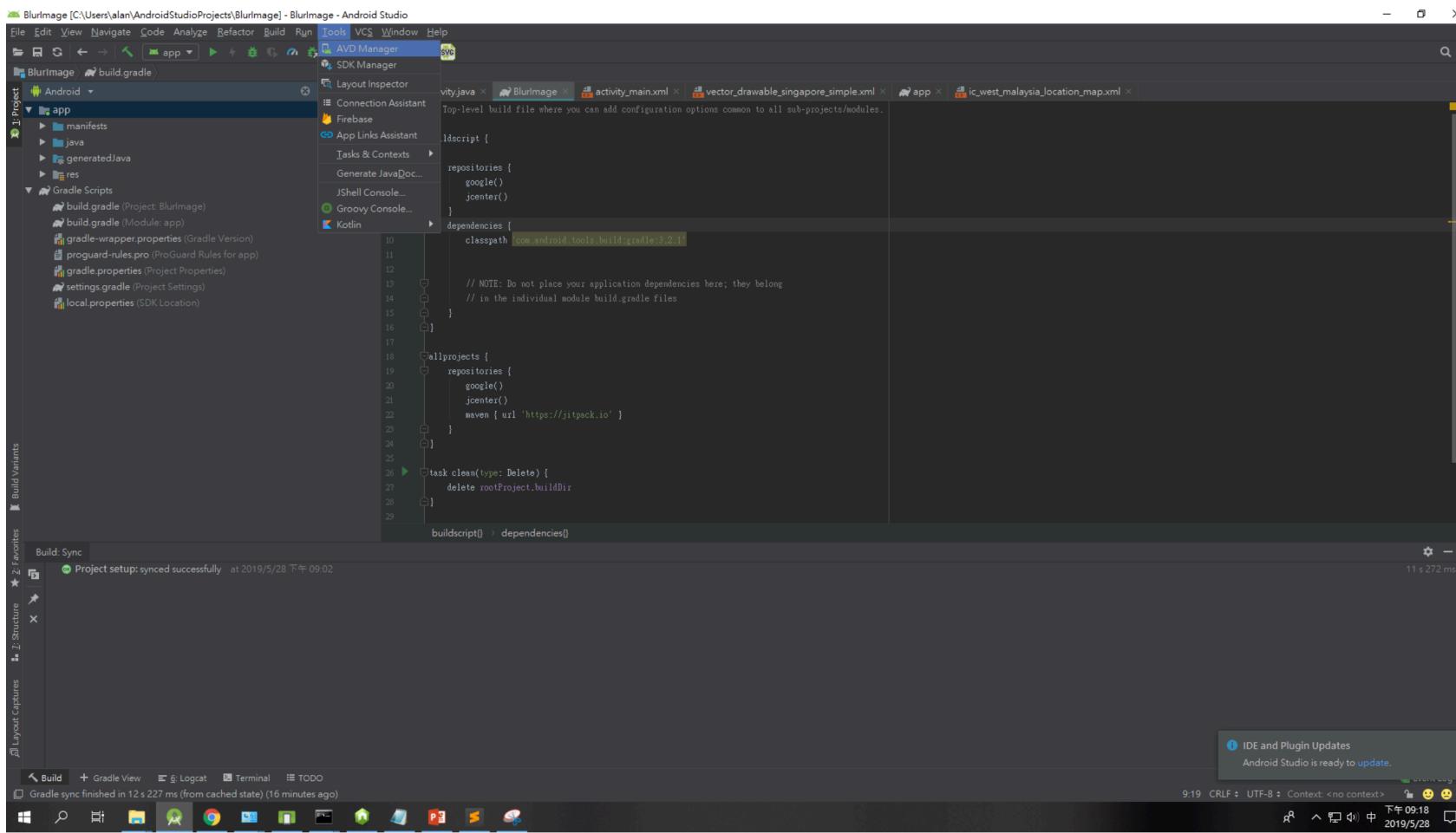


# React Native CLI - Android development environment

- Mac:
  - Add the following lines to your \$HOME/.bash\_profile or \$HOME/.bashrc config file:
  - ```
export ANDROID_HOME=$HOME/Library/Android/sdk
export PATH=$PATH:$ANDROID_HOME/emulator
export PATH=$PATH:$ANDROID_HOME/tools
export PATH=$PATH:$ANDROID_HOME/tools/bin
export PATH=$PATH:$ANDROID_HOME/platform-tools
```
  - `source $HOME/.bash_profile`

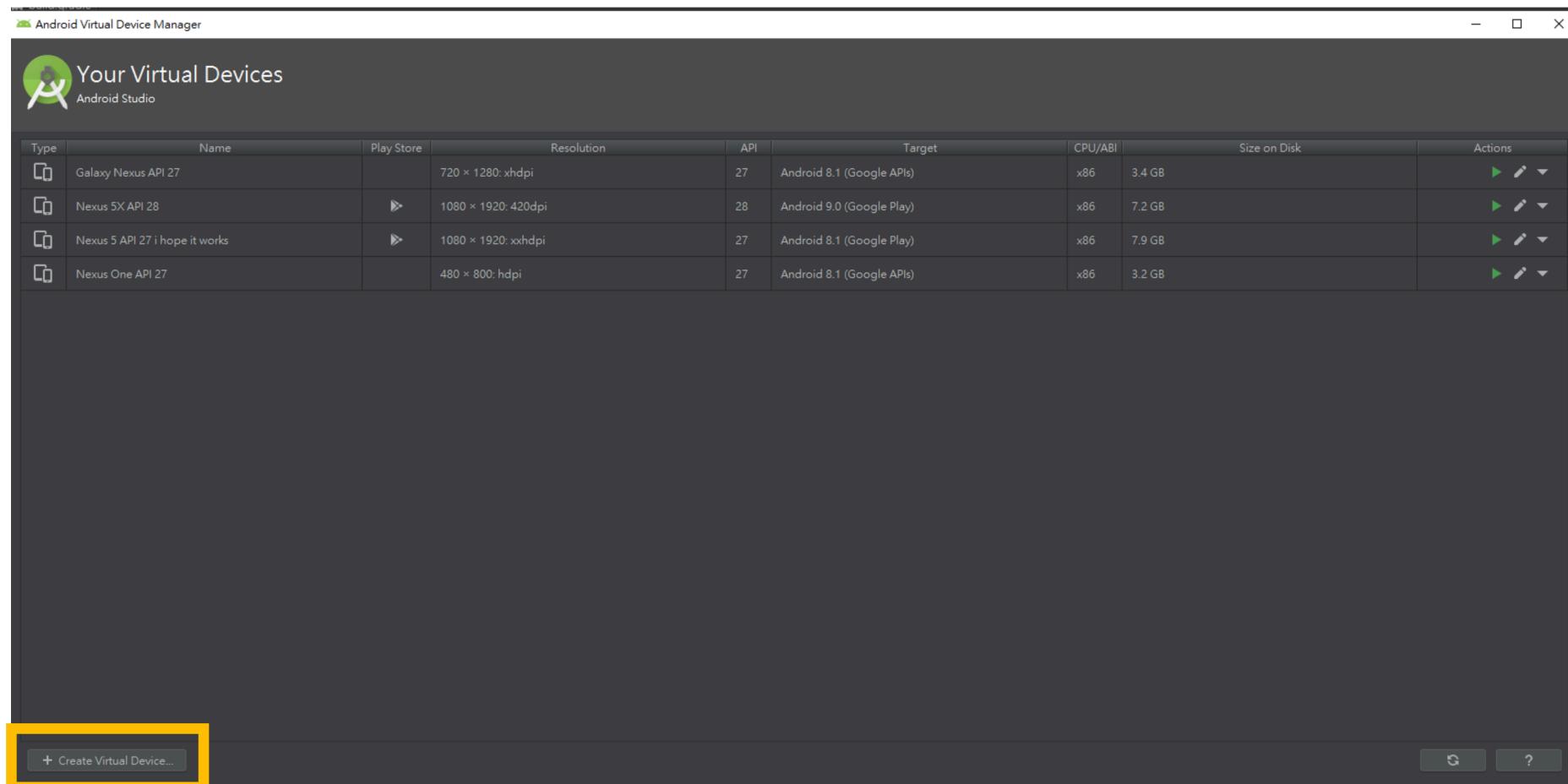
# Android Virtual Device

- Open AVD Manager



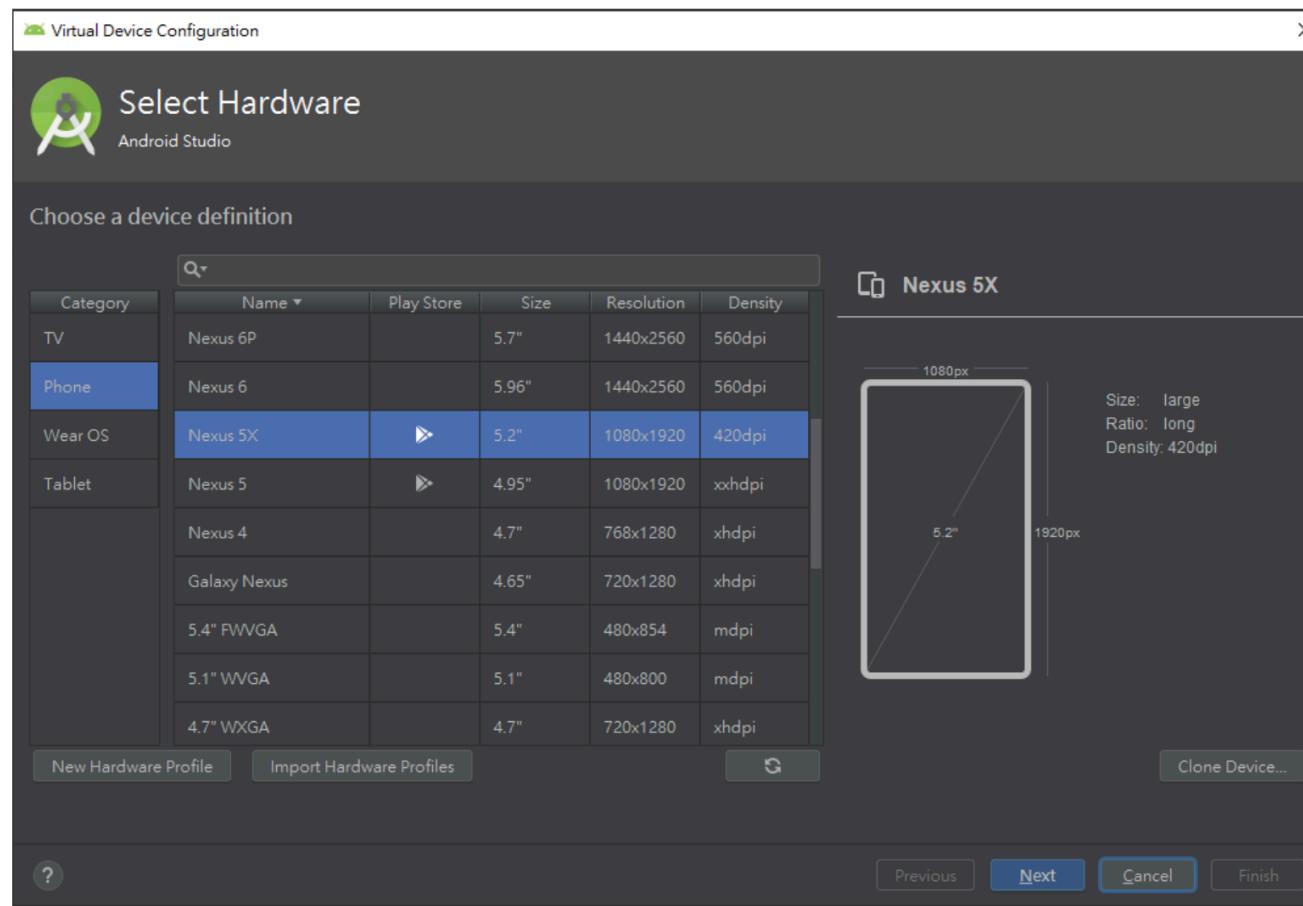
# Android Virtual Device

- Create Virtual Device



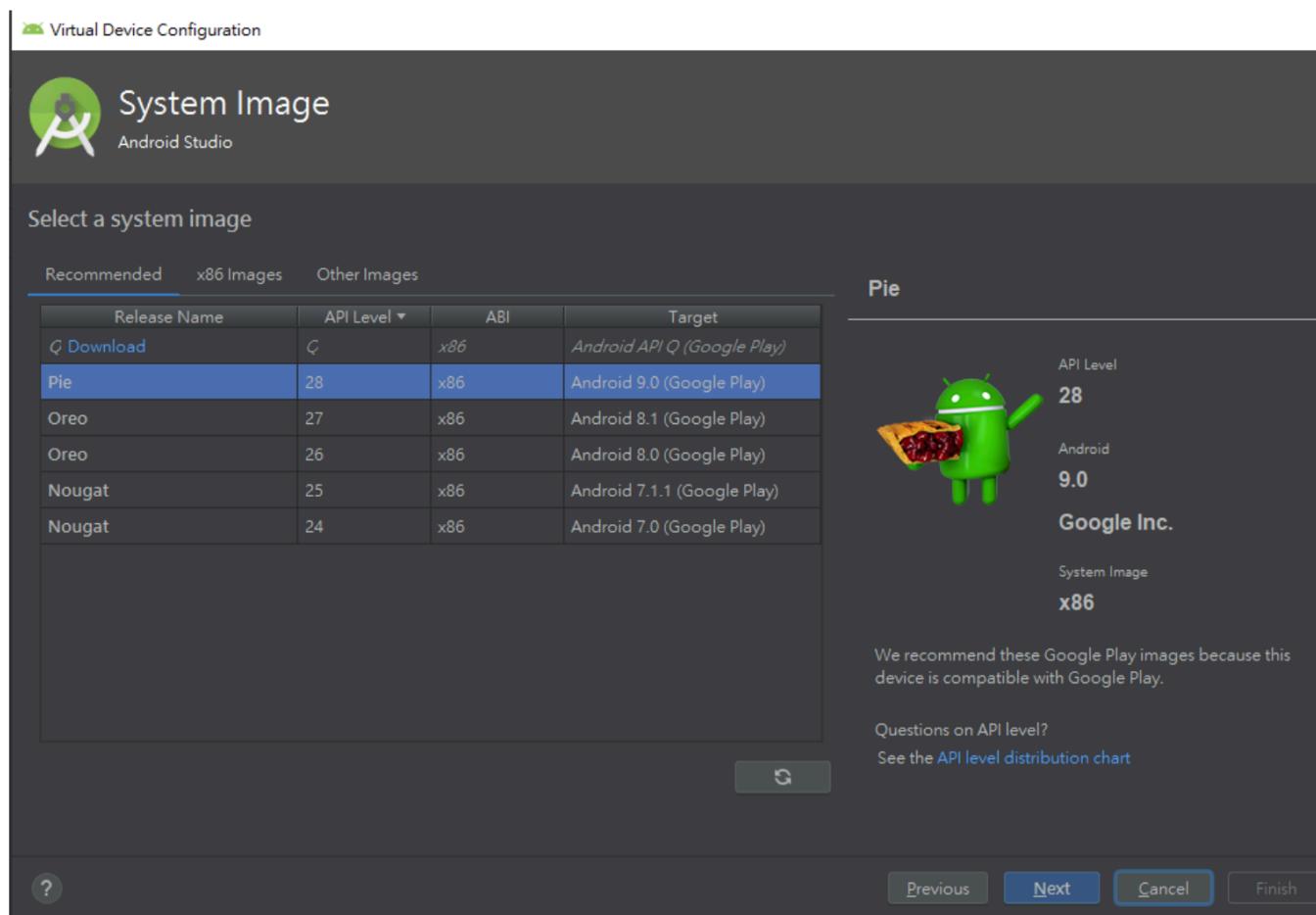
# Android Virtual Device

- Choose any phone you like



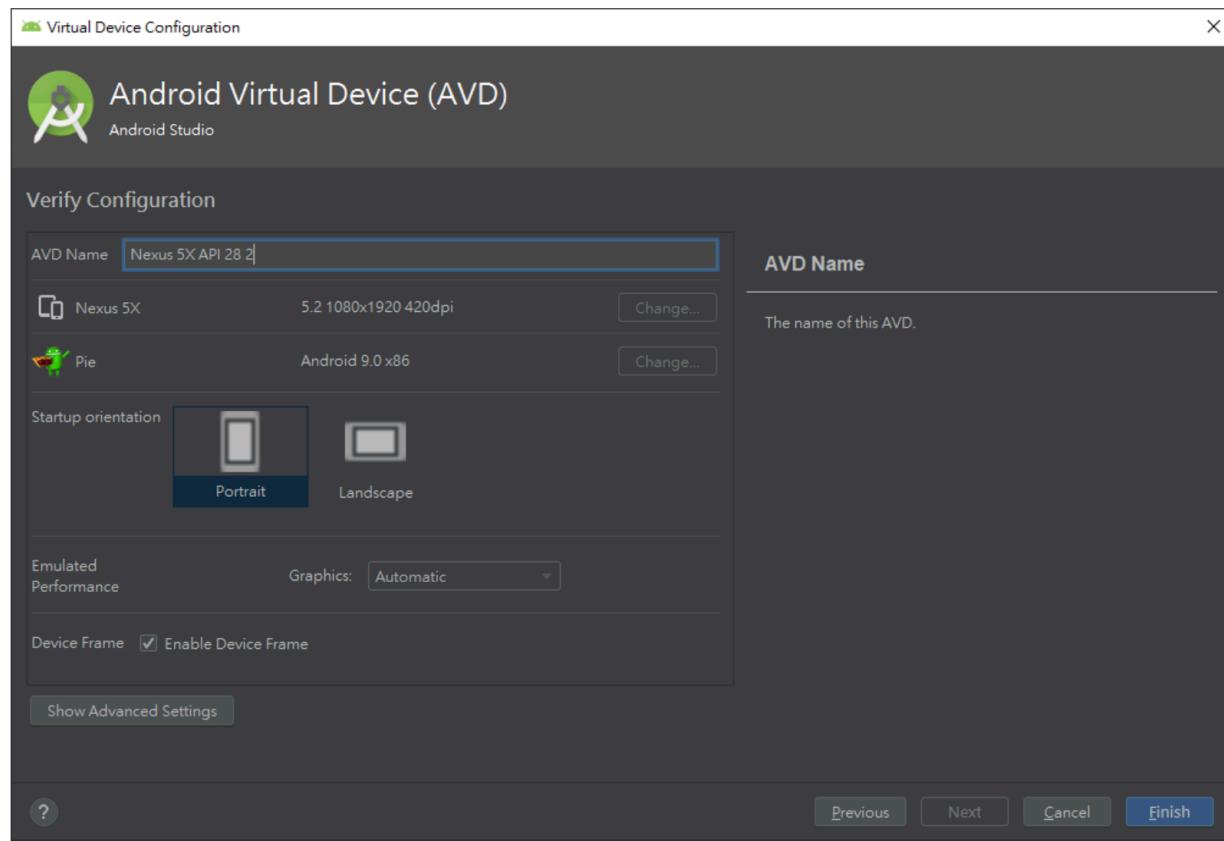
# Android Virtual Device

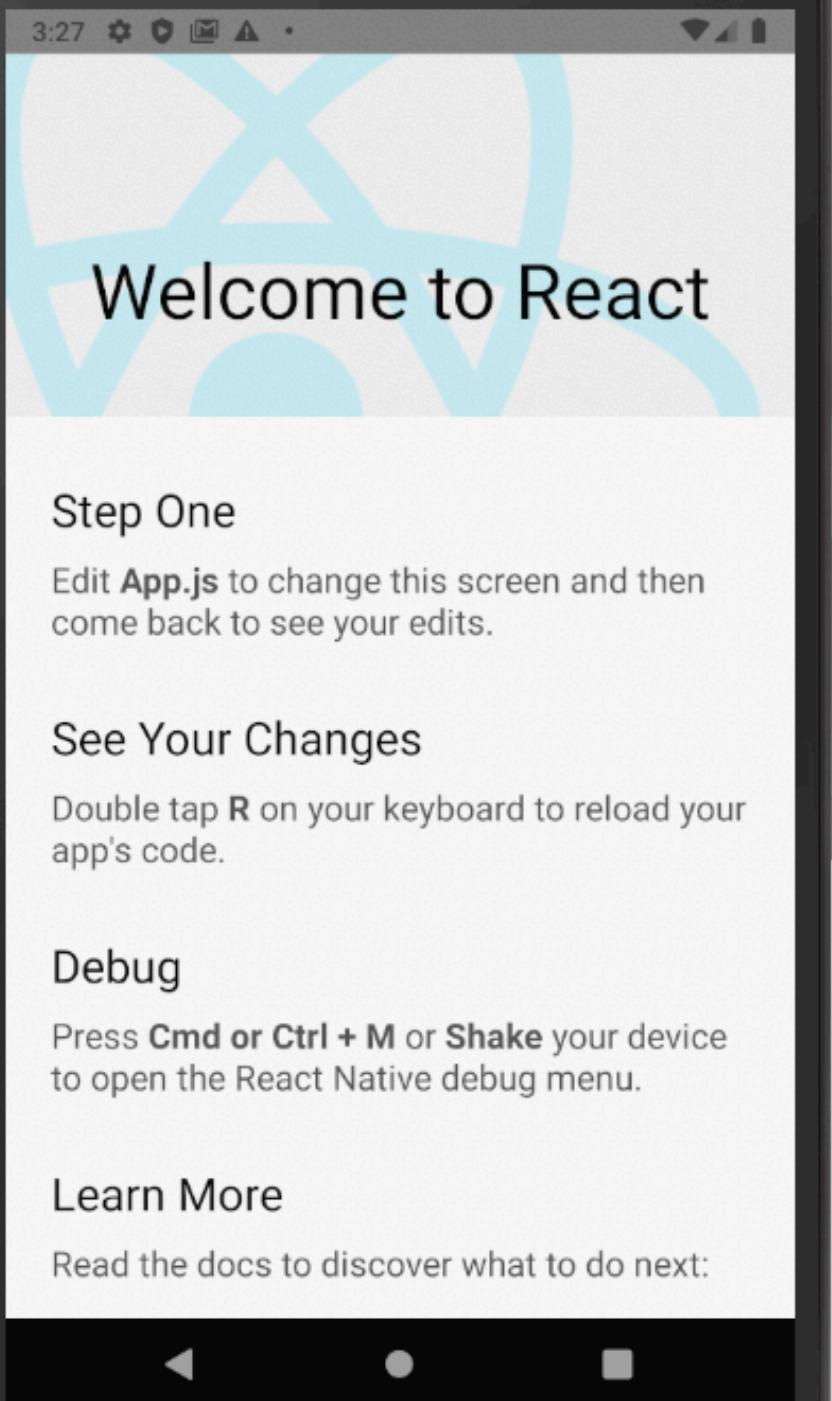
- Choose any system image you like



# Android Virtual Device

- Click finish and see the virtual Device





# React Native Project

---

- Creating a new application
  - `npx react-native init <project-name>`
- Preparing the Android device
- Running your React Native application
  - To start the application run **`npx react-native run-android`** inside your React Native project folder

# Expo CLI - Installation

- Requirements
  - [Node 12 LTS](#) or greater installed
- **npm install -g expo-cli**

[[source](#)]

# Expo CLI - React Native project

- **expo init <project\_name>**
- **cd <project\_name>**
- **npm start # you can also use: expo start**

```
→ AwesomeQuickProject git:(master) ✘ npm start
```

```
> @ start /Users/alanlin/Documents/software_studio_2020/lab/AwesomeQuickP
```

```
> expo start
```

```
Starting project at /Users/alanlin/Documents/software_studio_2020/lab/Awe  
Expo DevTools is running at http://localhost:19002  
Opening DevTools in the browser... (press shift-d to disable)  
Starting Metro Bundler on port 19001.
```

```
exp://192.168.50.177:19000
```



To run the app with live reloading, choose one of:

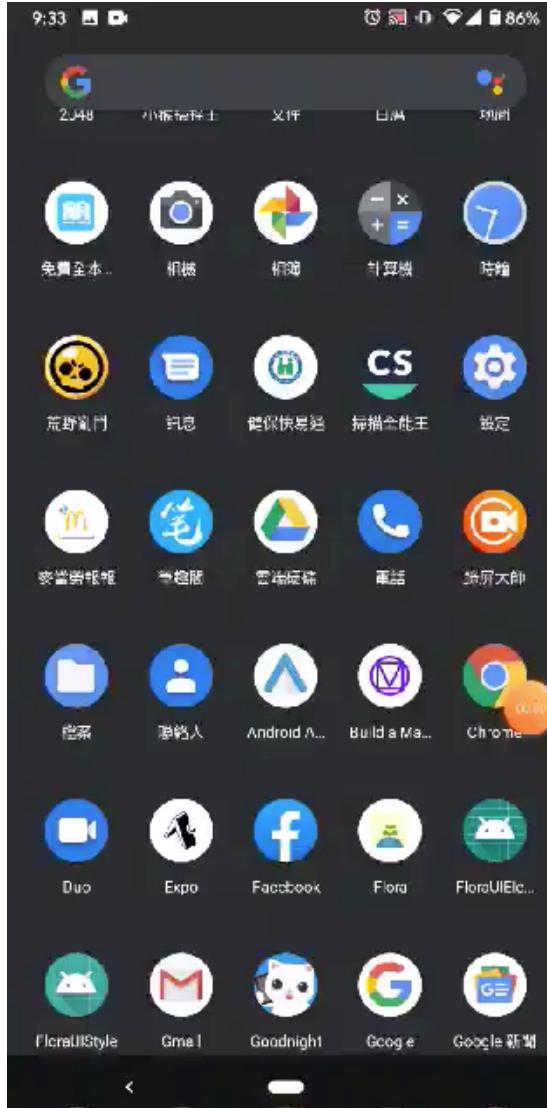
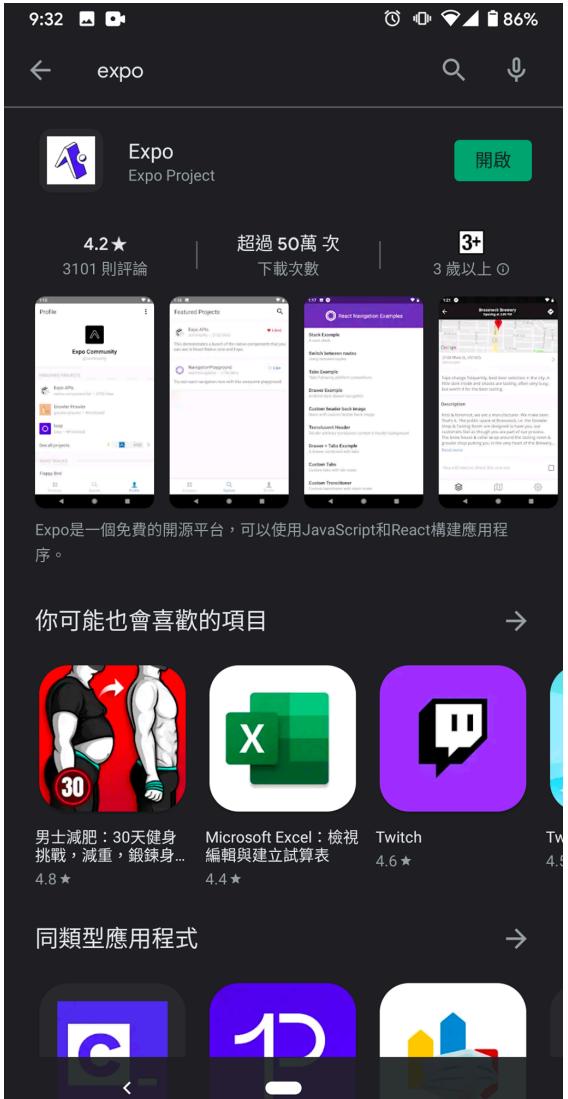
- Scan the QR code above with the Expo app (Android) or the Camera app
- Press **a** for Android emulator, or **i** for iOS simulator, or **w** to run on web
- Press **e** to send a link to your phone with email.
- Press **s** to sign in and enable more options.

# Expo CLI - Running your React Native application

---

- Install the [Expo](#) client app on your Android phone
- Connect to the same wireless network as your computer
- On Android, use the **Expo app to scan the QR code** from your terminal to open your project.

# Expo CLI - Running your React Native application



# React Native CLI vs Expo

|                      | React Native CLI                                                                                                                                                                                                                                      | Expo                                                                                                                                                                                                                              |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Advantages</b>    | <ul style="list-style-type: none"><li>• Add native modules written in Java/Objective-C</li><li>• Having control over the builds</li></ul>                                                                                                             | <ul style="list-style-type: none"><li>• Setting up a project is easy and can be done in minutes</li><li>• No build necessary to run the app</li></ul>                                                                             |
| <b>Disadvantages</b> | <ul style="list-style-type: none"><li>• Needs Android Studio and XCode to run the projects</li><li>• You can't develop for iOS without having a mac</li><li>• Setting up a working project properly is rather complicated and can take time</li></ul> | <ul style="list-style-type: none"><li>• You can't add native modules</li><li>• You can't use libraries that use native code in Objective-C/Java</li><li>• Big app size(The standard Hello World app is about 25MB big )</li></ul> |

Happy Coding!!