

ANDROID

Mobile device programming

Introduction to android

ANDROID

- ระบบปฏิบัติการสำหรับอุปกรณ์พกพา เช่น โทรศัพท์มือถือ แท็บเล็ตคอมพิวเตอร์ เน็ตบุ๊ก ทำงานบนลินุกซ์ เคอร์เนล เริ่มพัฒนาโดยบริษัทแอนดรอยด์ จากนั้นบริษัทแอนดรอยด์ถูกซื้อโดยกูเกิล และนำแอนดรอยด์ไปพัฒนาต่อ
- ภายหลังถูกพัฒนาในนามของ Open Handset Alliance
- ทางกูเกิลได้เปิดให้นักพัฒนาสามารถแก้ไขโค้ดต่างๆ ด้วยภาษาจาวา และควบคุมอุปกรณ์ผ่านทางชุด Java libraries ที่กูเกิลพัฒนาขึ้นโดยแอนดรอยด์ (Android) ถูกตั้งชื่อเลียนแบบหุ่นยนต์ในเรื่อง สตาร์วอร์ส ที่ชื่อดรอยด์ ซึ่งเป็นหุ่นยนต์ที่สร้างขึ้นมาเลียนแบบมนุษย์เป็นซอฟต์แวร์ระบบปฏิบัติการที่มีโครงสร้างแบบเรียงกับซ้อนหรือแบบสแต็ก (Stack) โดยใช้ลินุกซ์ เคอร์เนล(Linux Kernel) เป็นพื้นฐานของระบบ และใช้ภาษา Java ในการพัฒนา มี Android SDK เป็นเครื่องมือสำหรับการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ อีกทีหนึ่ง โดยระบบปฏิบัติการ แอนดรอยด์ เริ่มพัฒนาเมื่อปี พ.ศ. 2550 โดยบริษัทแอนดรอยด์ร่วมกับ Google จากนั้นเมื่อปี พ.ศ.2550 ได้มีการร่วมมือกันกว่า 30 บริษัทชั้นนำเพื่อพัฒนาระบบ



Software installation

- Android studio [<https://developer.android.com/studio>]
- Git SCM [<https://git-scm.com/>]
- Virtual Box [<https://www.virtualbox.org/>]
- Genymotion [<https://www.genymotion.com/>]
- AdobeXD [<https://www.adobe.com/sea/products/xd.html>]
- FileZilla [<https://filezilla-project.org/>]

Android Studio

- Android Studio เป็น IDE Tool จาก Google ໄ้พัฒนา Android
- สำหรับ Android Studio เป็น IDE Tools ไล่สุดจาก Google ໄ้พัฒนาโปรแกรม Android โดยเฉพาะ โดยพัฒนาจากแนวคิดพื้นฐานมาจาก IntelliJ IDEA คล้าย ๆ กับการทำงานของ Eclipse และ Android ADT Plugin
- โดยวัตถุประสงค์ของ Android Studio คือต้องการพัฒนาเครื่องมือ IDE ที่สามารถพัฒนา App บน Android ใ้มีประสิทธิภาพมากขึ้น ทั้งด้านการออกแบบ GUI ที่ช่วยให้สามารถ Preview ตัว App มุมมองที่แตกต่างกันบน Smart Phone แต่ละรุ่น
- สามารถแสดงผลบางอย่างได้ทันทีโดยไม่ต้องทำการรัน App บน Emulator รวมทั้งยังแก้ไขปรับปรุงในเรื่องของความเร็วของ Emulator ที่ยังเจอปัญหากันอยู่ในปัจจุบัน

Android SDK & API Level

Android Software Development Kit (Android SDK) เปรียบเสมือน Library ที่ใช้ในการพัฒนา Application สำหรับ Android เนื่องจากตัว Android มีหลายเวอร์ชันและแต่ละเวอร์ชันมี Feature, GUI ที่ไม่เหมือนกันทำให้เกิด Android SDK ออกมาหลายเวอร์ชันให้เลือกใช้งาน

API Level หมายถึงเวอร์ชันของ API ที่ให้นักพัฒนานำมาใช้งานใน SDK เวอร์ชันนั้นๆ เช่น มือถือของผมใช้ Android 8 ก็จะสามารถใช้งาน API Level 26 ซึ่งอาจจะมีการเพิ่ม Feature ใหม่ๆ เพิ่มขึ้นมา ทำให้มือถือที่ใช้ Android เวอร์ชันเก่ากว่านี้จะไม่สามารถใช้งาน API Level นี้ได้


วิธีการติดตั้ง sdk

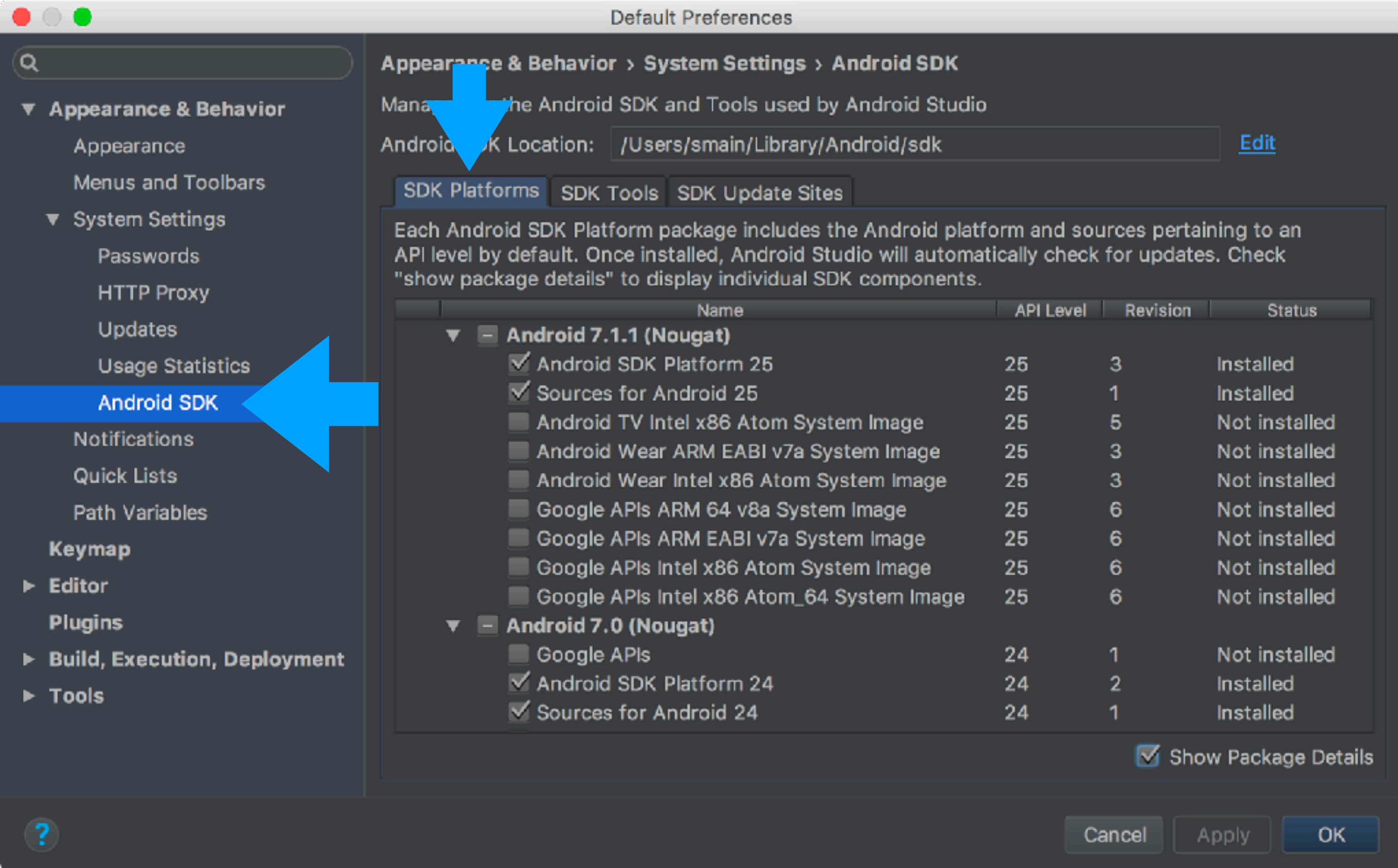


Android Studio

Version 3.1.3

- 🌟 Start a new Android Studio project
- ⚡ Start a new Flutter project
- 📁 Open an existing Android Studio project
- ⬇ Check out project from Version Control ▾
- 📱 Profile or debug APK
- 📂 Import project (Gradle, Eclipse ADT, etc.)
- 📄 Import an Android code sample

- 
- ⚙ Configure ▾
 - Get Help ▾
 - SDK Manager**
 - Preferences
 - Plugins
 - Import Settings
 - Export Settings
 - Settings Repository...
 - Check for Updates



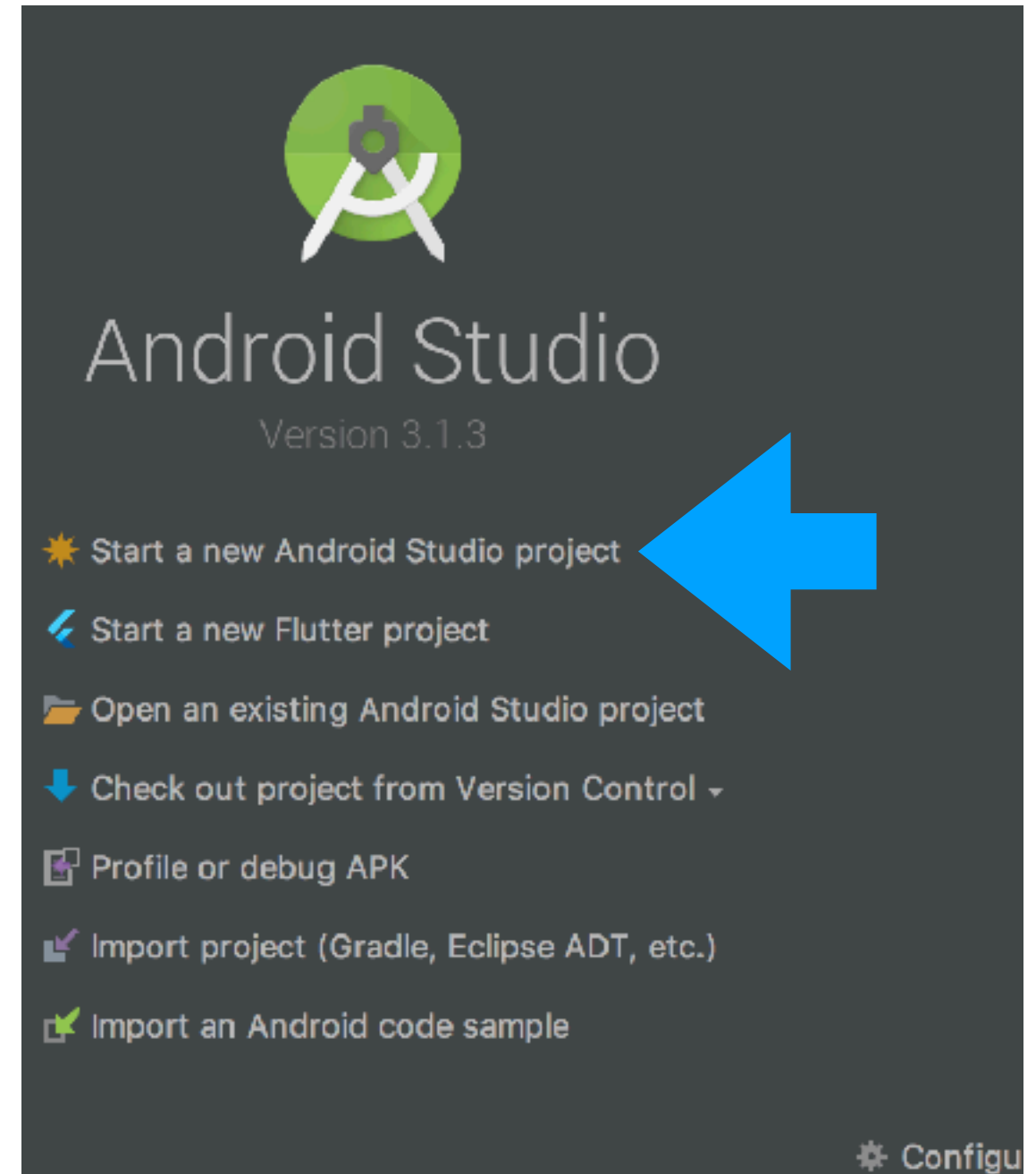
Check box ของ sdk version ที่ต้องการแล้วกดปุ่ม OK ด้านล่าง

วิธีการใช้งาน

android studio เบื้องต้น

วิธีการสร้าง Project ใหม่

- กดเลือก Start a new Android Studio project



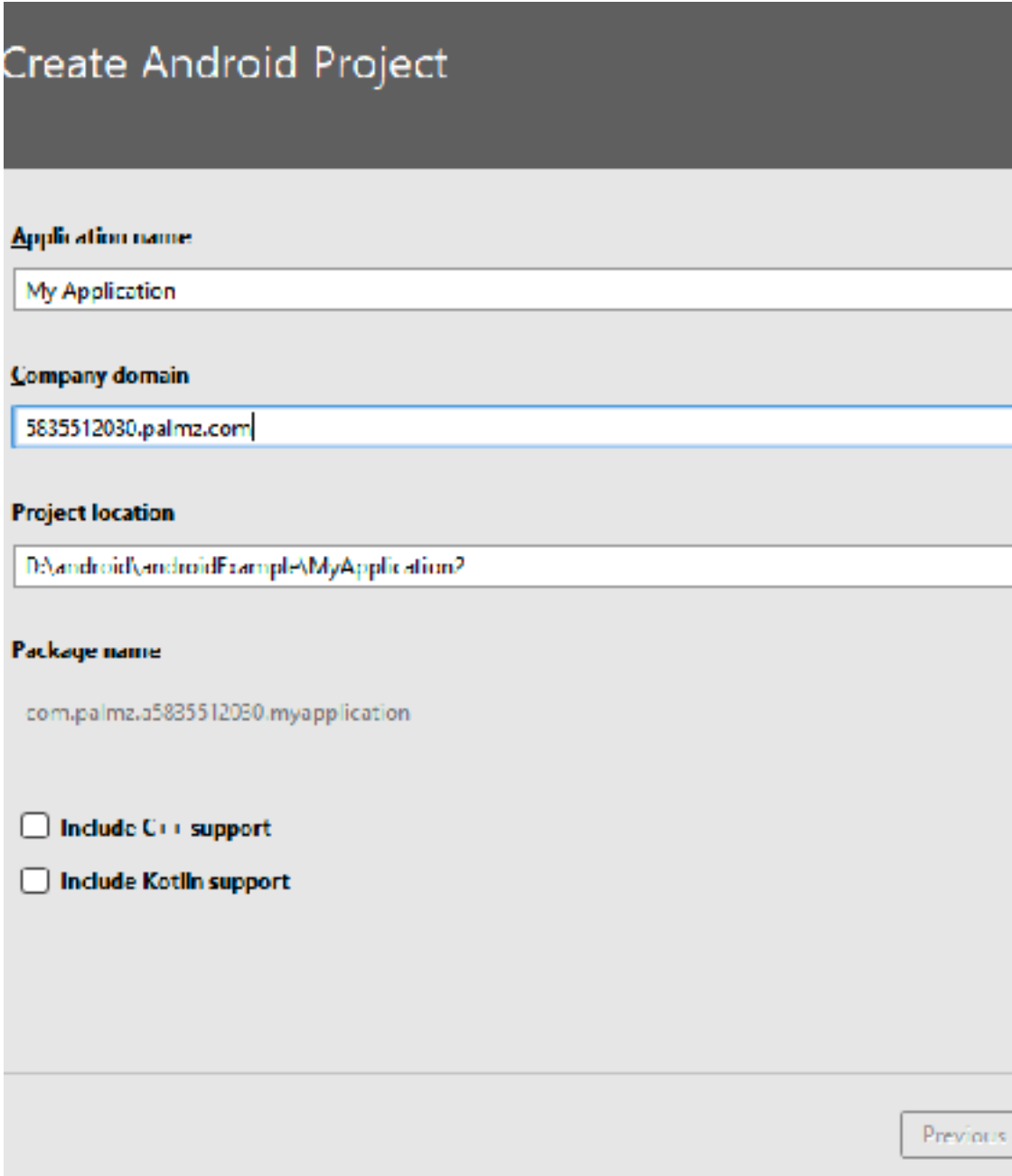
วิธีการใส่ข้อมูลตอน new Project

Application Name — ใส่ชื่อแอปพลิเคชันที่เราต้องการ ต้องขึ้นต้นด้วยตัวใหญ่

Company Domain — ใส่โดเมนอะไรก็ได้ของเรา (ตรงนี้สำคัญเพราะถ้าหากซ้ำกับของผู้อื่นจะไม่สามารถนำลง Play Store ได้)

Project Location — ได้เรียกทอี่ที่ต้องการเก็บไฟล์ไว้

Include C++/Kotlin Support — ทำให้โปรเจกต์ของเราสามารถใช้งานภาษา C++, Kotlin



The screenshot shows the 'Create Android Project' dialog box. It has a dark header with the title 'Create Android Project'. Below the header, there are four input fields and two checkboxes. The first field is 'Application name' with the value 'My Application'. The second field is 'Company domain' with the value '5835512030.palmz.com'. The third field is 'Project location' with the value 'D:\android\androidExample\MyApplication?'. The fourth field is 'Package name' with the value 'com.palmz.a5835512030.myapplication'. Below these fields are two checkboxes: 'Include C++ support' and 'Include Kotlin support', both of which are unchecked. At the bottom right, there is a 'Previous' button.

Field	Value
Application name	My Application
Company domain	5835512030.palmz.com
Project location	D:\android\androidExample\MyApplication?
Package name	com.palmz.a5835512030.myapplication

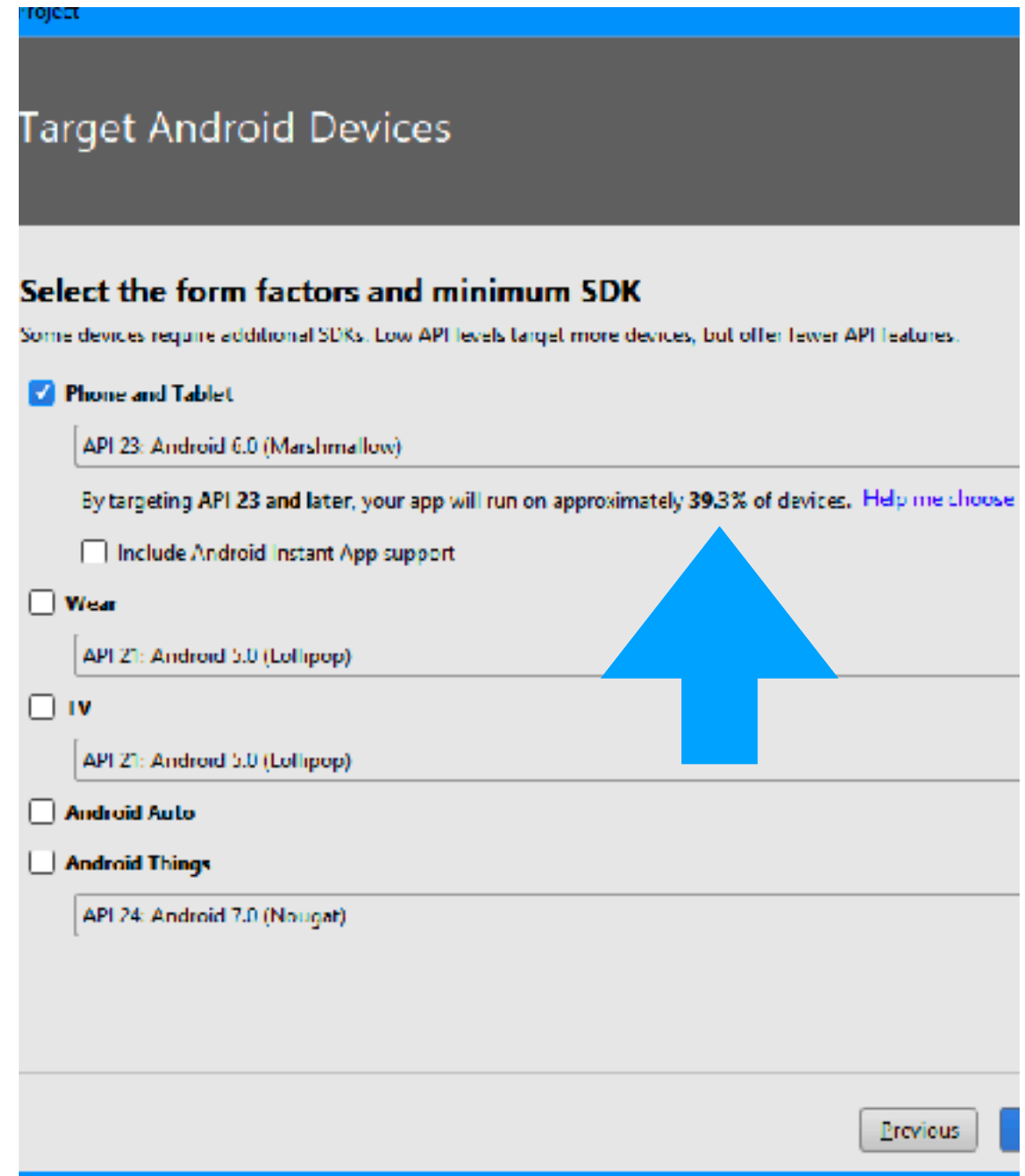
☐ Include C++ support

☐ Include Kotlin support

Previous

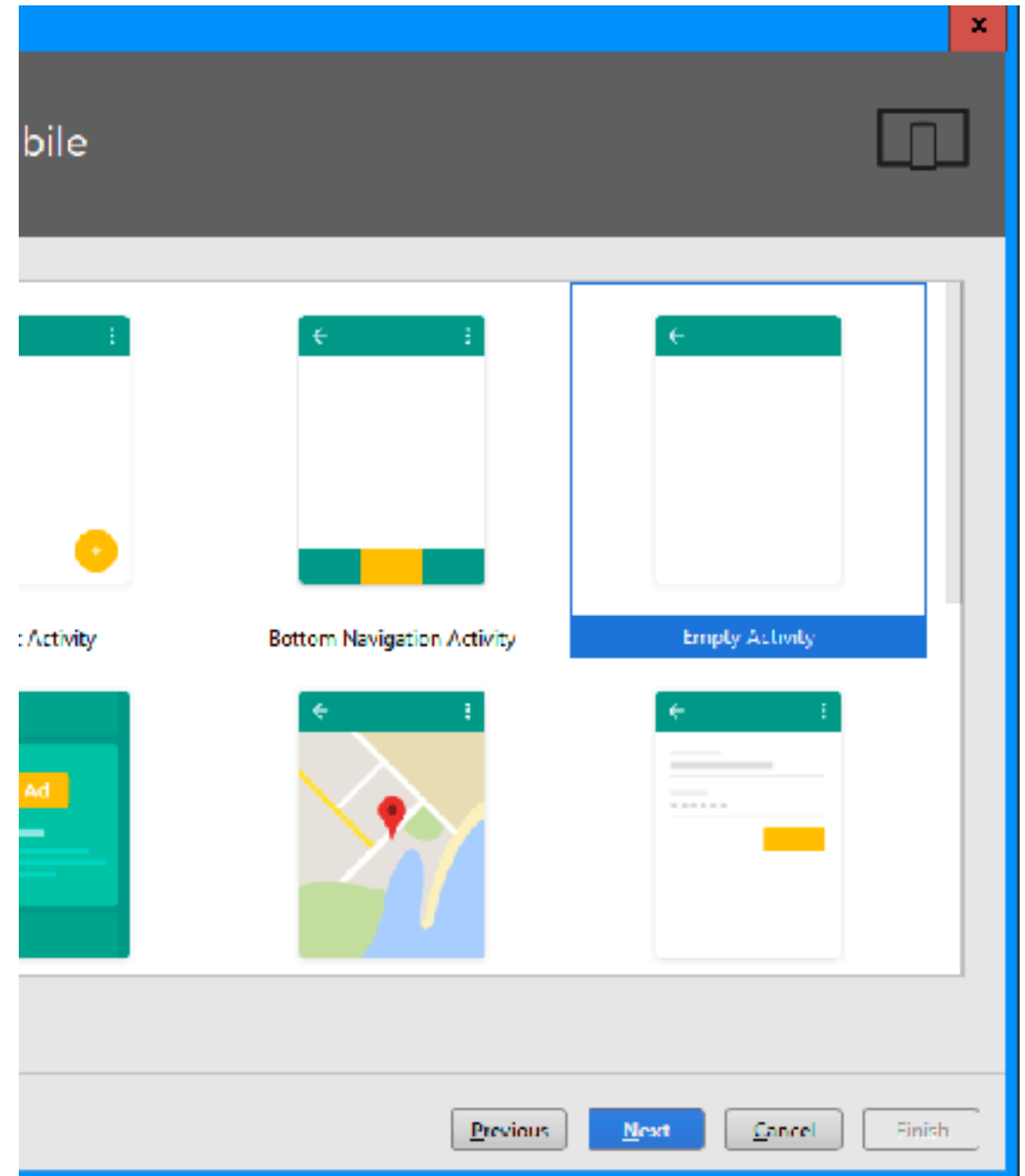
เลือกแพลตฟอร์ม

- เลือกแพลตฟอร์มที่เราต้องการให้แอปพลิเคชันของเรารันได้ ในที่นี้ขอเลือกเป็น Phone and Tablet เนื่องจากจะเน้นใช้งานแค่ในมือถือและแท็บเล็ต เลือก Android API เวอร์ชัน 23 สำหรับใช้กับ Android 6.0 (Marshmallow) เนื่องจากเป็นเวอร์ชันที่ค่อนข้างเสถียร
- หลังจากเลือก Android API แล้วด้านล่างจะมีบอกว่าจำนวนผู้ใช้งาน Android ที่เปอร์เซ็นต์จากทั้งหมดบนโลกที่กำลังใช้งาน Android Version นี้และสามารถรันแอปนี้ได้ (ตรงลูกศร)



เลือก initial template

- เลือกรูปแบบ Activity ที่ต้องการใช้งาน (Activity ก็เปรียบเสมือน Class GUI ของแอปของเรา)
- ให้เลือก Empty Activity



Configure Activity

ใส่ชื่อ Activity, ชื่อ Layout หลักของ
แอป (ตรงนี้ปล่อยเป็นค่า Default ไว้
ได้)

Project

Configure Activity

Creates a new empty activity

←

Activity Name

MainActivity

☒ Generate Layout File

Layout Name

activity_main

☒ Backwards Compatibility (AppCompat)

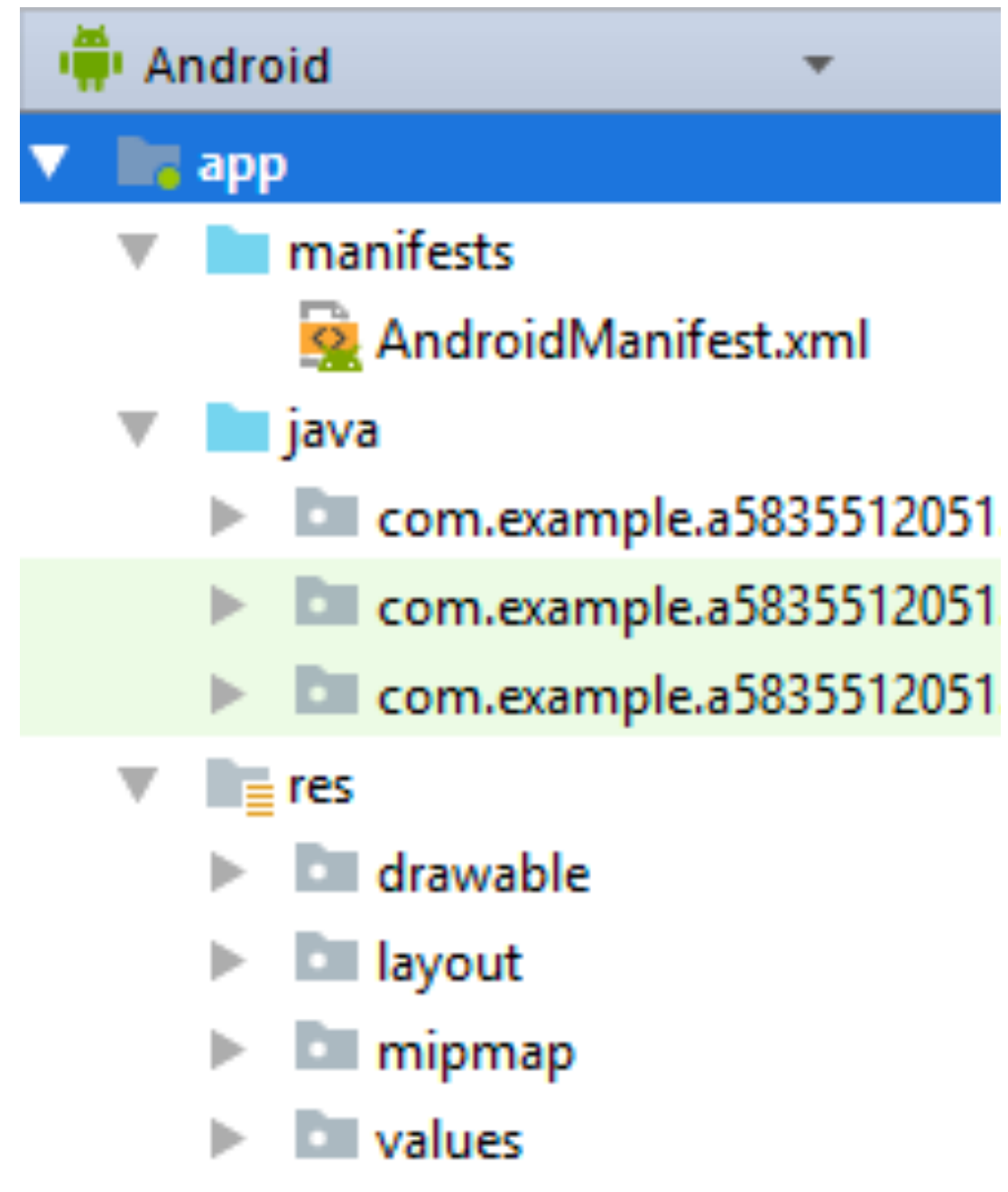
Previous

Project Directory

manifests—สำหรับเก็บไฟล์.xml ที่ใช้สำหรับตั้งค่าต่างๆ ของแอปของเรา เช่น ชื่อแอป, ชื่อ Activity หลักที่จะให้เรียกใช้งานตอนรับแอป, Permission ต่างๆ ของแอปของเรา

java — สำหรับเก็บไฟล์.java ของแอป เป็นส่วน Controller หลักของแอปที่ใช้ในการประมวลผล

res — สำหรับเก็บไฟล์.xml เพื่อใช้สร้าง GUI ของแอป และเก็บ Resource ต่างๆ เช่น รูปภาพ, สีธีม เป็นต้น



Menu

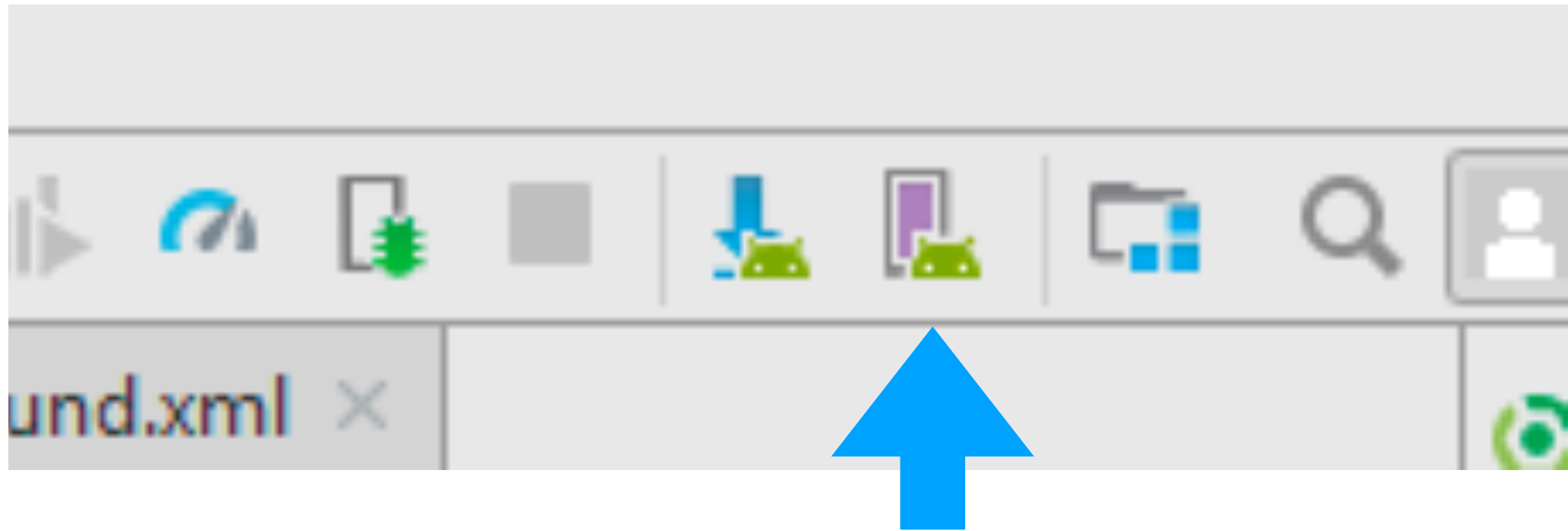


- Make Project (รูปก้อนสี่เหลี่ยม) สำหรับ Compile แอปของเรา
- Run (รูปสามเหลี่ยมสีเขียว) สำหรับใช้ Compile แอปของเราและรันบน Emulator
- Apply (รูปสายฟ้าสีเหลือง) สำหรับใช้ Compile แอปและอัปเดตแอปบน Emulator (ในกรณีที่แอปเรารันบน Emulator อยู่แล้วให้ใช้ปุ่มนี้แทนปุ่ม Run จะเร็วกว่า)
- SDK Manager (รูปลูกศรชี้ลงสีฟ้า) สำหรับใช้จัดการเวอร์ชัน SDK ในเครื่อง
- AVD Manager (รูปโทรศัพท์สีม่วง) สำหรับใช้จัดการ Emulator ในเครื่อง

วิธีการรับ android

- มีอยู่ด้วยกันทั้งหมด 2 วิธี 3 รูปแบบ ดังนี้
 - วิธีที่ 1 รับผ่าน virtual machine มี 2 แบบคือ
 - ใช้ Android Virtual Device (AVD) ค่อนข้างช้าและใช้ทรัพยากรเครื่องเยอะ
 - ใช้ Genymotion เร็วกว่า avd แต่ก็ยังใช้ทรัพยากรเครื่องเยอะอยู่
 - วิธีที่ 2 รับผ่านเครื่องจริง โดยต้องมีกระบวนการดังนี้
 - เครื่องมือที่ใช้ระบบปฏิบัติการ android
 - เปิด developer mode
 - ต่อสาย ota (สาย usb ที่มาพร้อมกับเครื่อง ถ้าเป็นแบบอื่นอาจจะใช้ไม่ได้)

วิธีการสร้าง emulator (AVD)



กดที่ปุ่มตามลูกศร

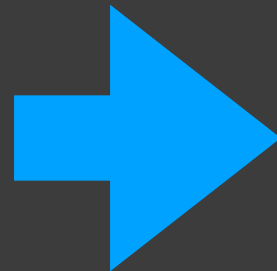
เพื่อเข้าสู่ Virtual Device Configuration

Your Virtual Devices

Android Studio



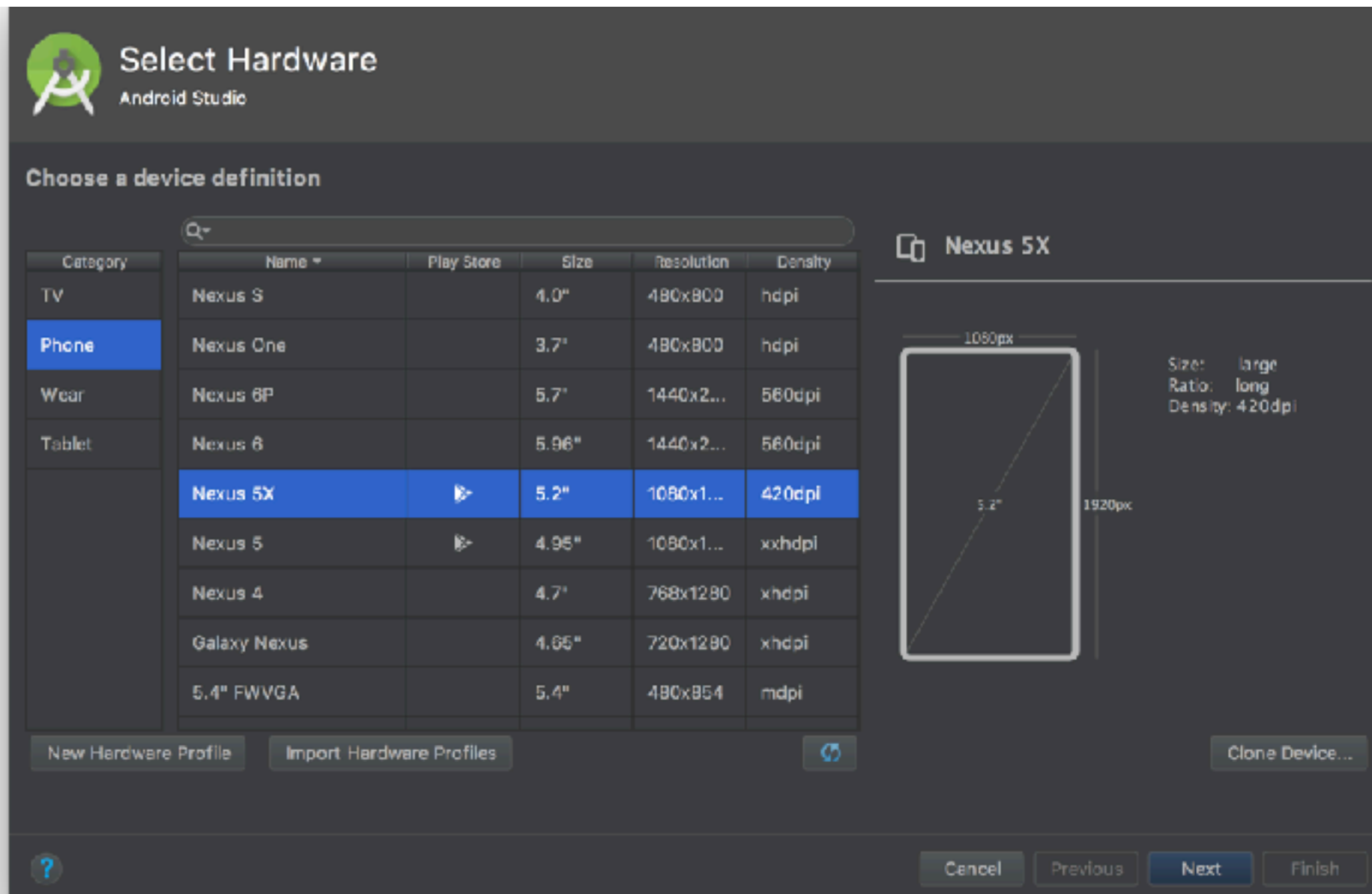
Virtual devices allow you to test your application without having to own the physical devices.



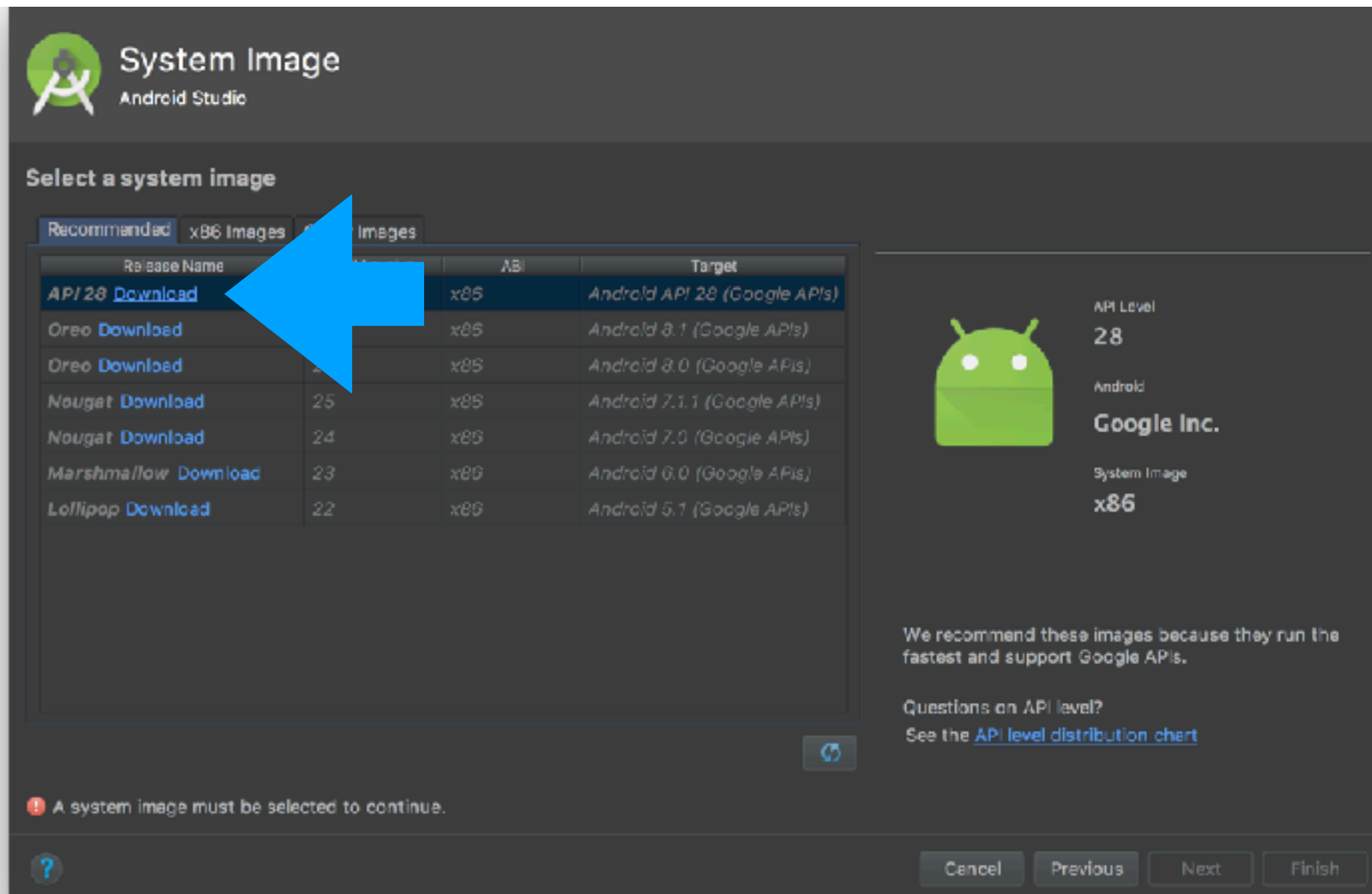
+ Create Virtual Device...

To prioritize which devices to test your application on, visit the [Android Dashboards](#), where you can get up-to-date information on which devices are active in the Android and Google Play ecosystem.

กดที่ปุ่มตามลูกศร

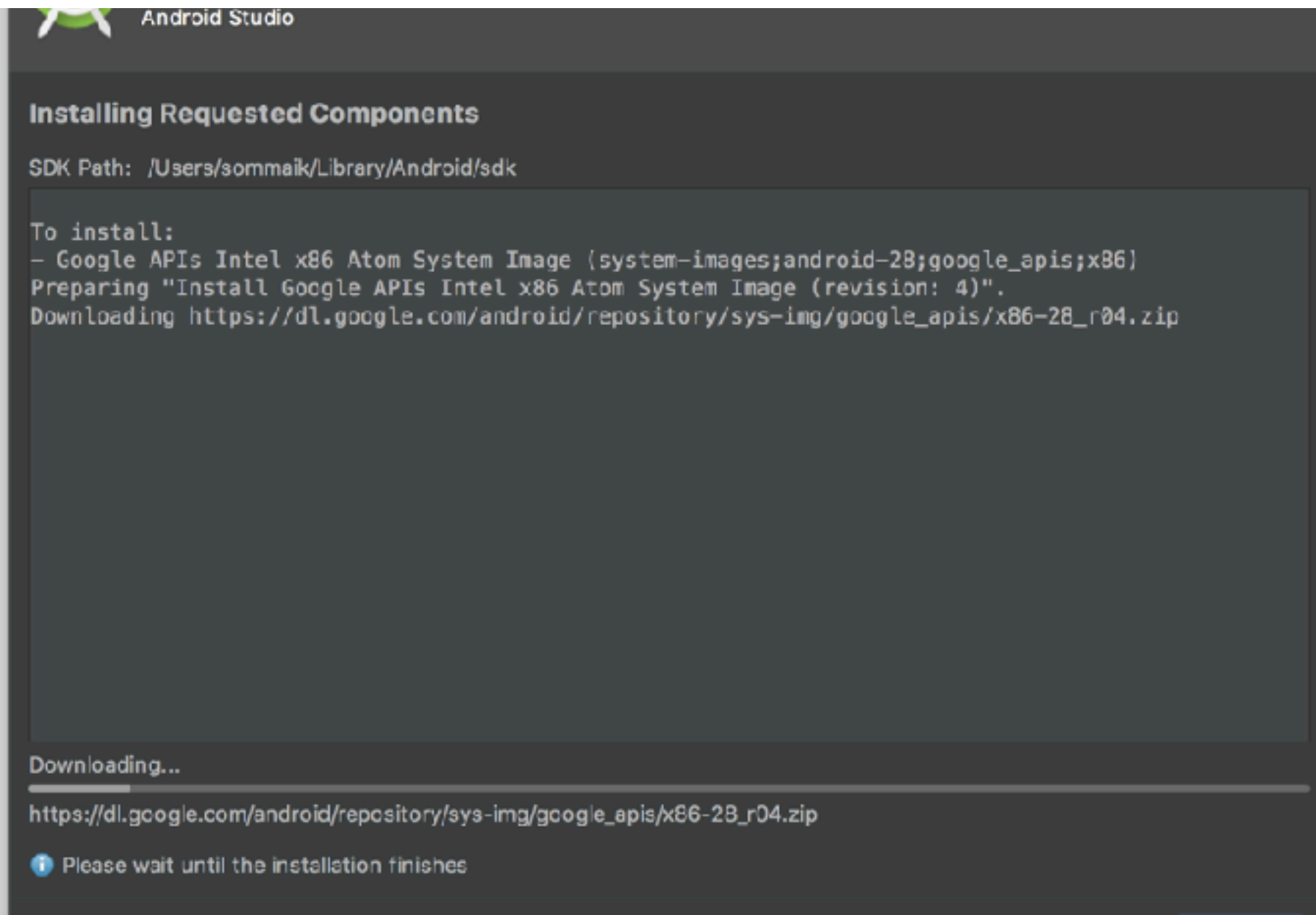


เลือก device ที่ต้องการ
จากนั้นให้กด Next



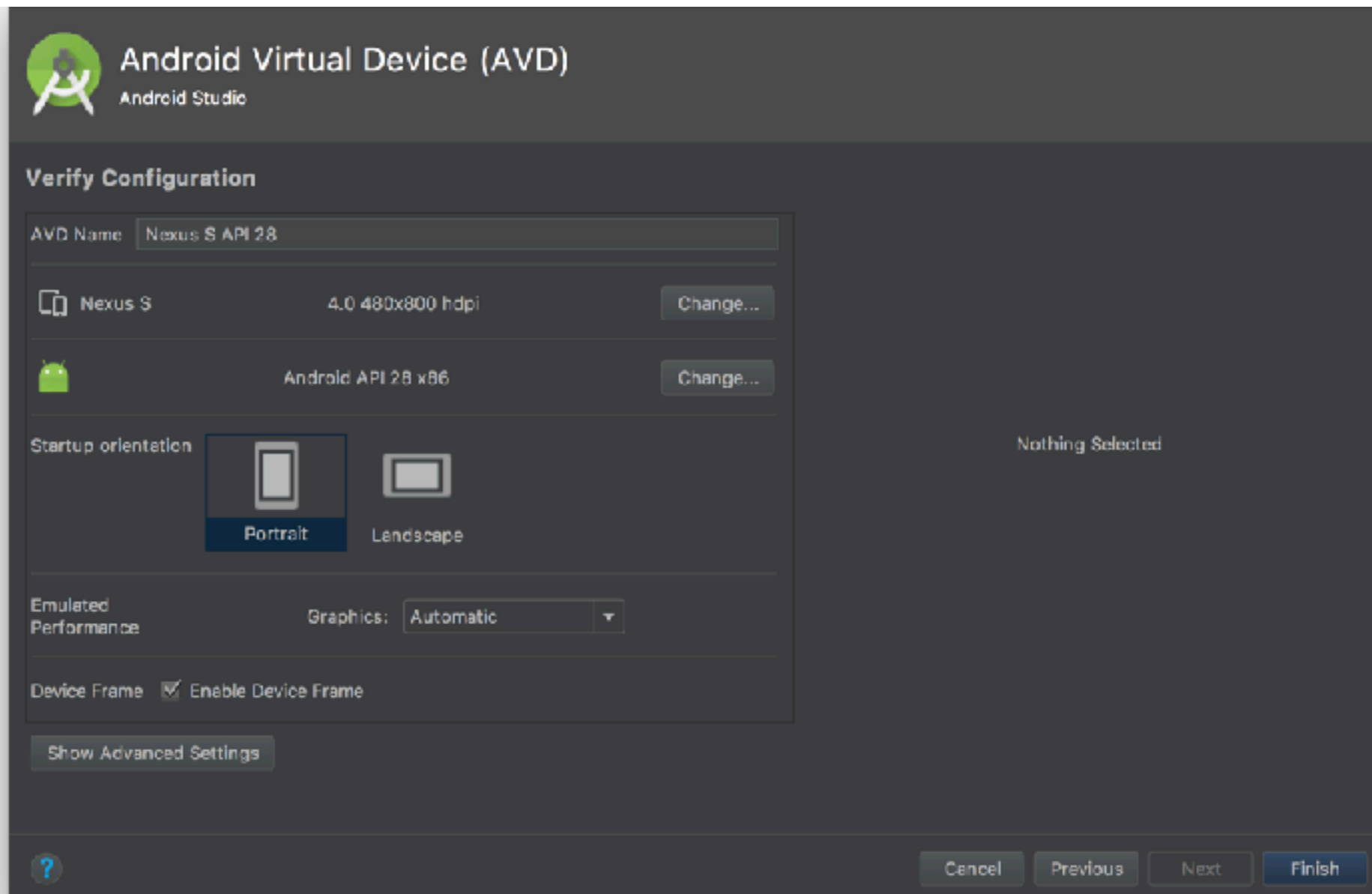
กดที่ Download

เพื่อ load image



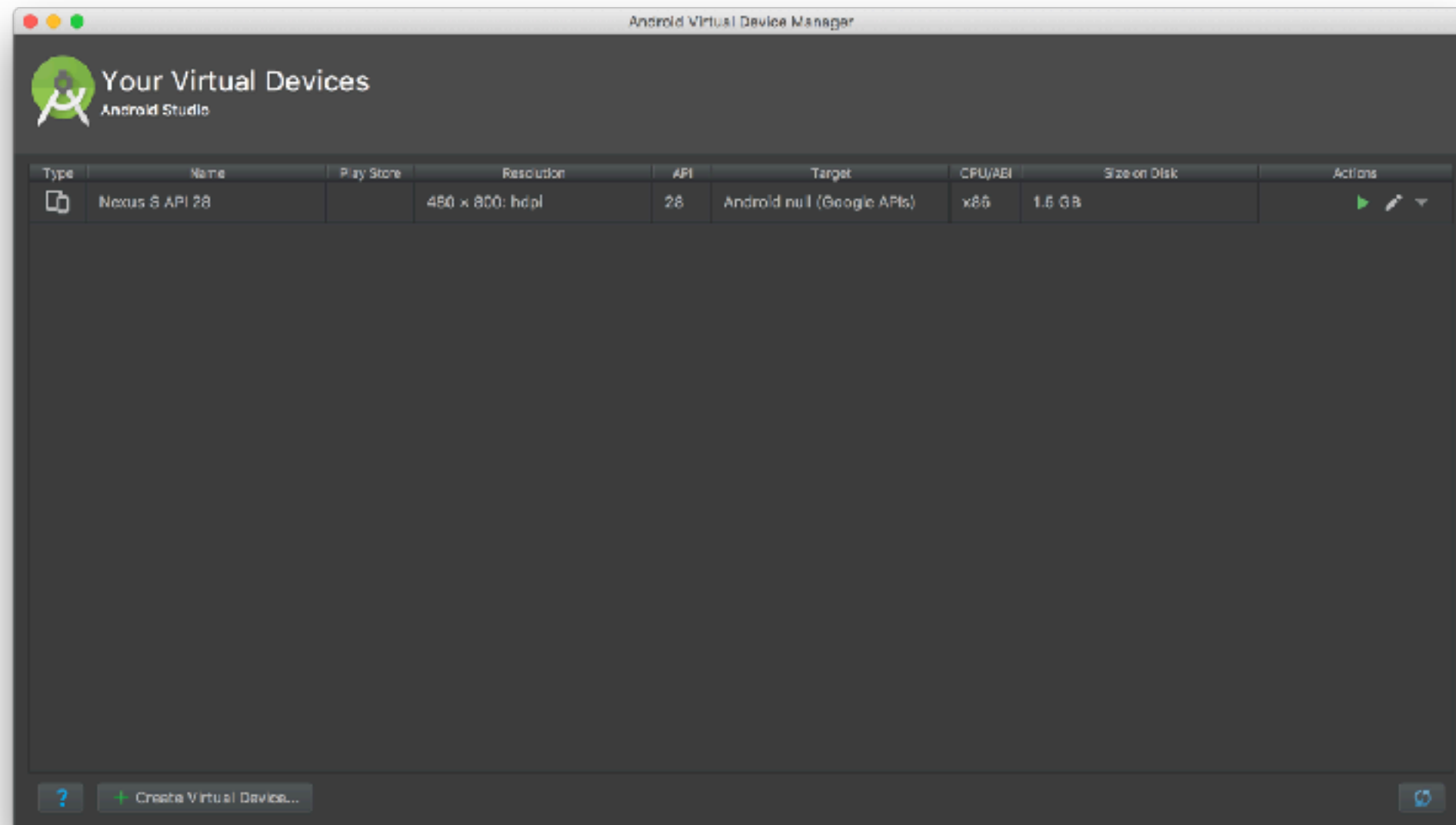
ให้รอนกว่าจะ download เสร็จ

เมื่อ download เสร็จให้กด finish



ตั้งค่าของ avd

ในกรณีนี้ให้ใช้ค่า default ไป และกด Finish



เมื่อเสร็จแล้ว

จะได้ผลลัพธ์ดังภาพ

วิธีการติดตั้ง
Genymotion



Register

Username*

Usage type*

Email address*

Company type*

Password*

Country

เข้าไป create user ตาม link

<https://www.genymotion.com/account/create/>

Other downloads available

System Requirements

Microsoft Windows 7, 8/8.1, 10 (32/64 bit)
64 bit CPU, with VT-x or AMD-V capability, enabled in BIOS settings
Recent and dedicated GPU
400 MB disk space
2GB RAM

[Checksum Windows \(with VirtualBox\)](#)
[Checksum Windows \(without VirtualBox\)](#)

with Virtualbox

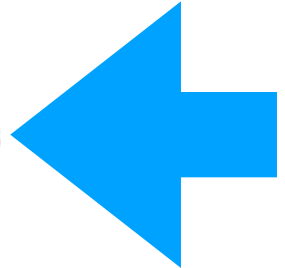
[Download for Windows](#)

154MB

without Virtualbox

[Download for Windows](#)

48MB



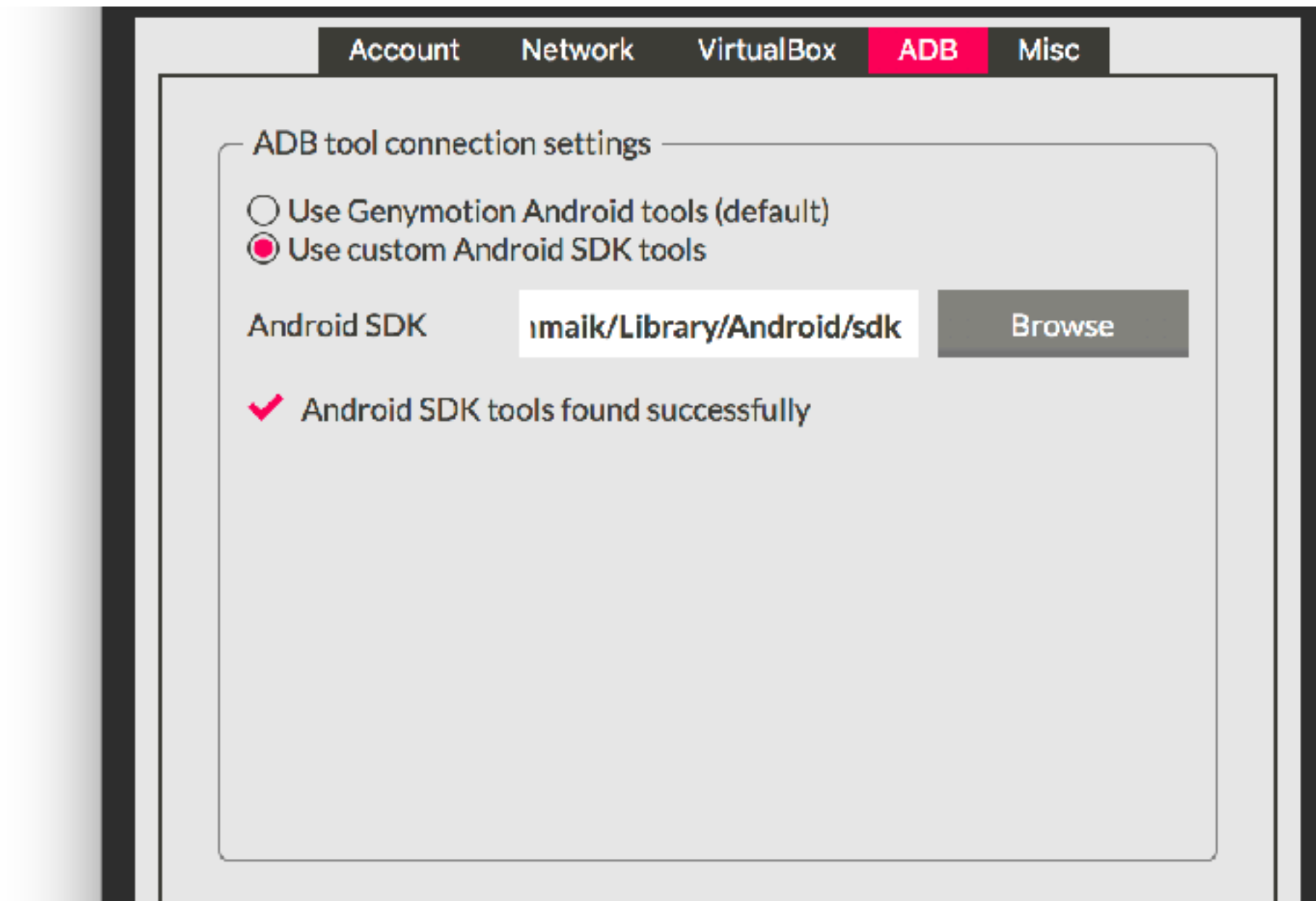
System Requirements

Ubuntu 18.04 (Bionic beaver) - 64bit or above

Ubuntu 18.04, Debian 9, Fedora 27 (64bit)

เมื่อ Register เสร็จให้เข้า url เพื่อ download

<https://www.genymotion.com/download/>



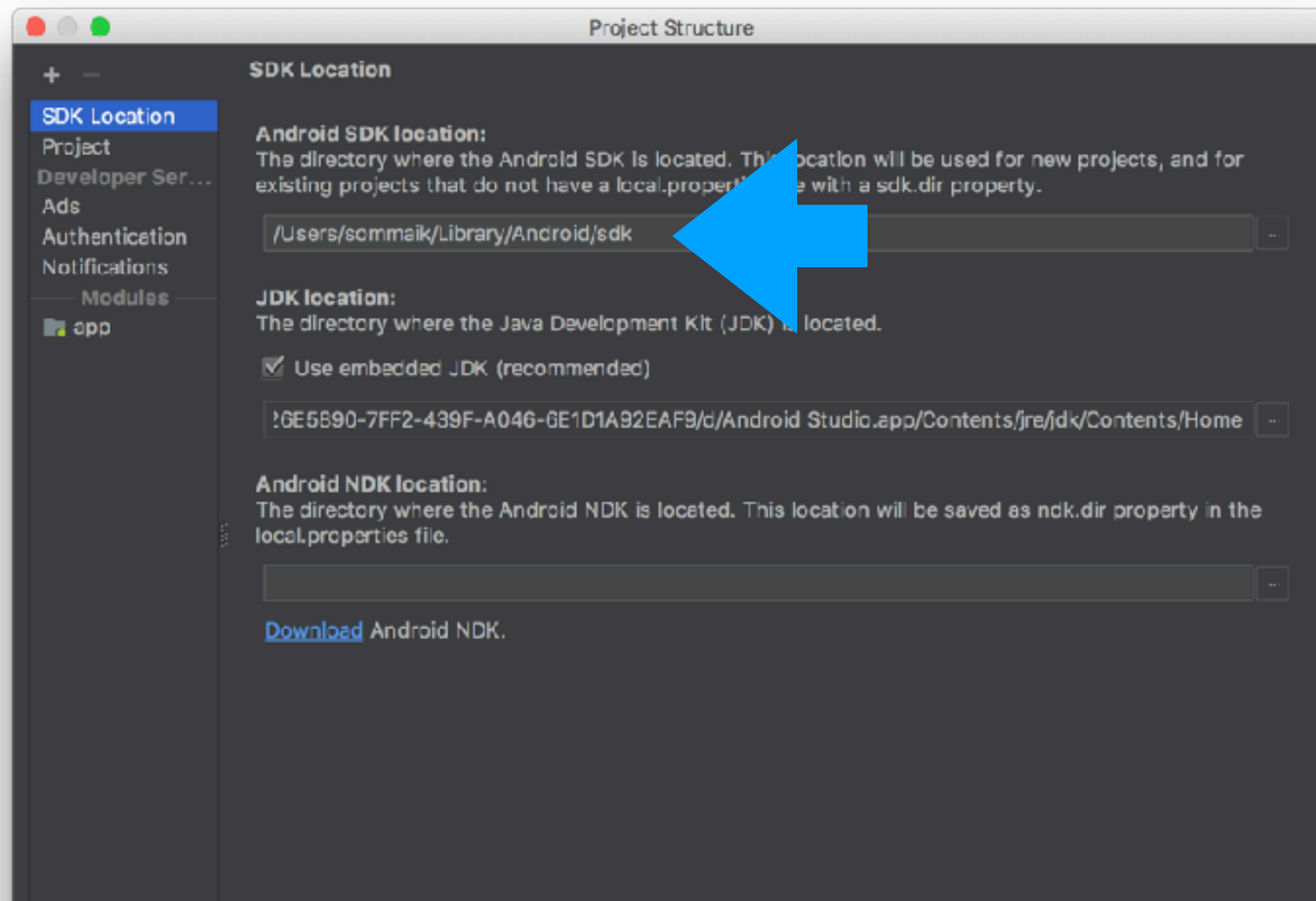
ตั้งค่า ADB

เลือกเป็น Use custom Android SDK tools

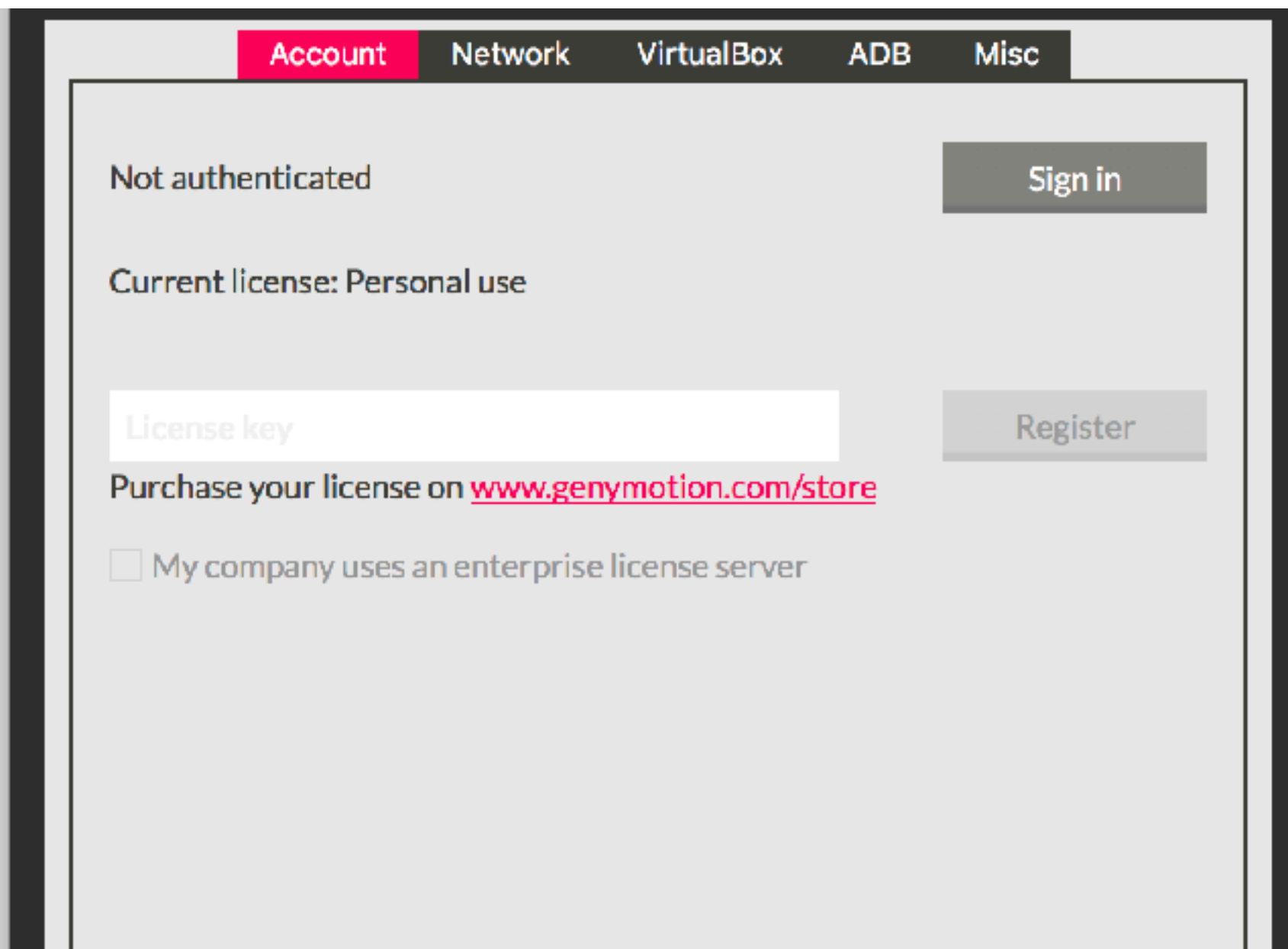


กดที่ปุ่มตามลูกศร

เพื่อเข้าสู่ Project Structure

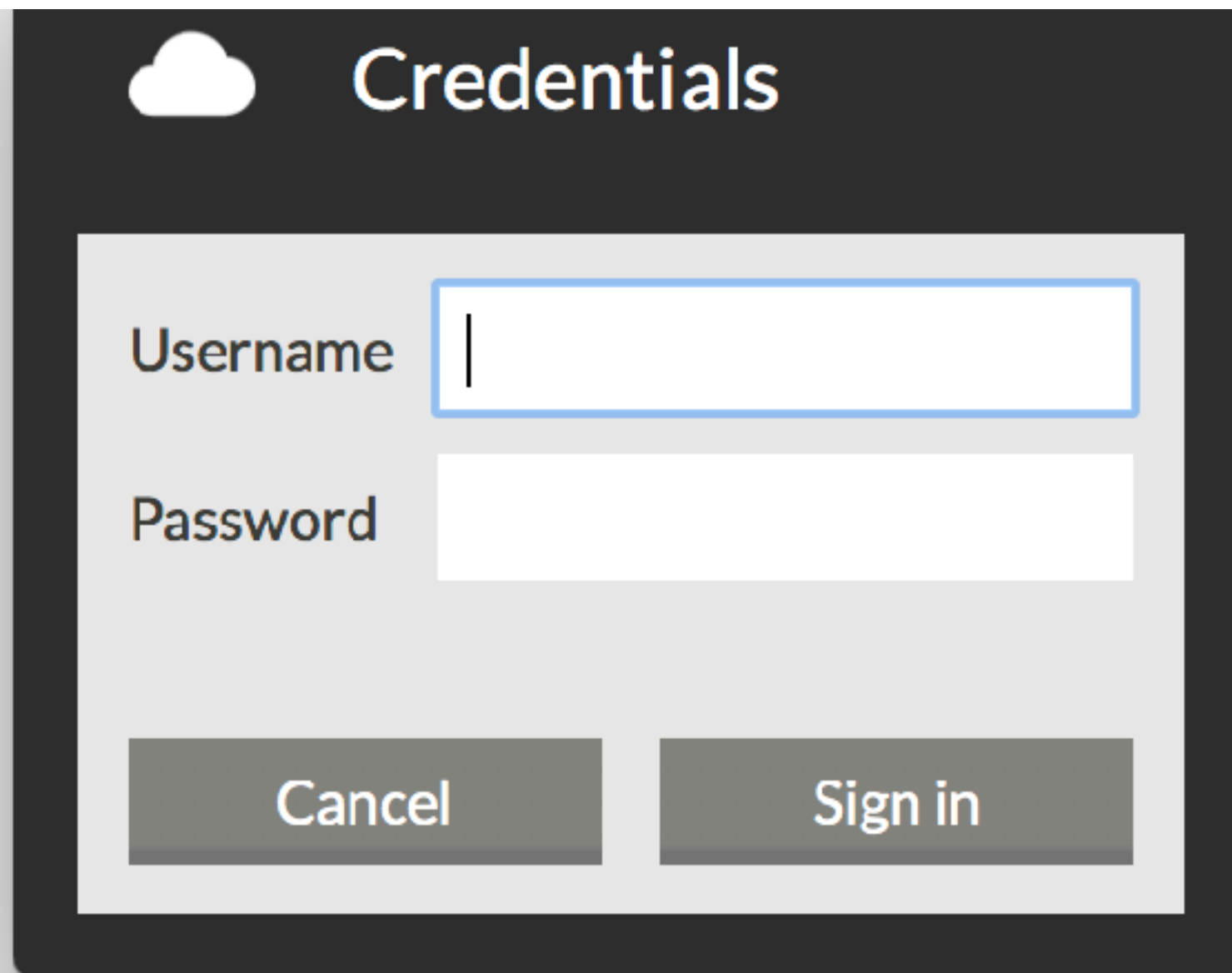


นำค่าในช่อง Android SDK Location
ไปใส่ใน program Genymotion



เลือก Tab Account

แล้วกดปุ่ม Sign in เพื่อเข้าสู่ระบบ



Cloud icon

Credentials

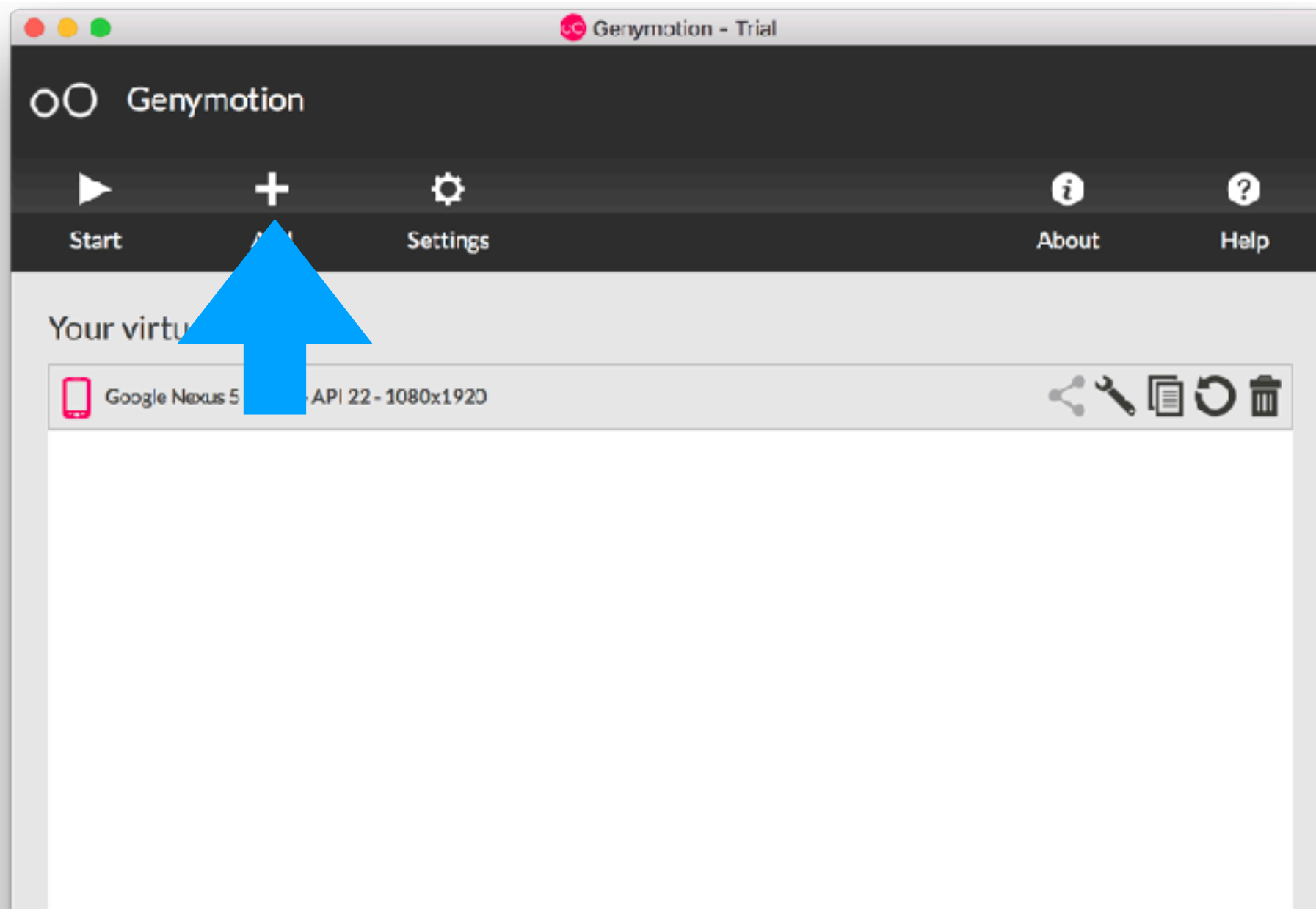
Username

Password

Cancel Sign in

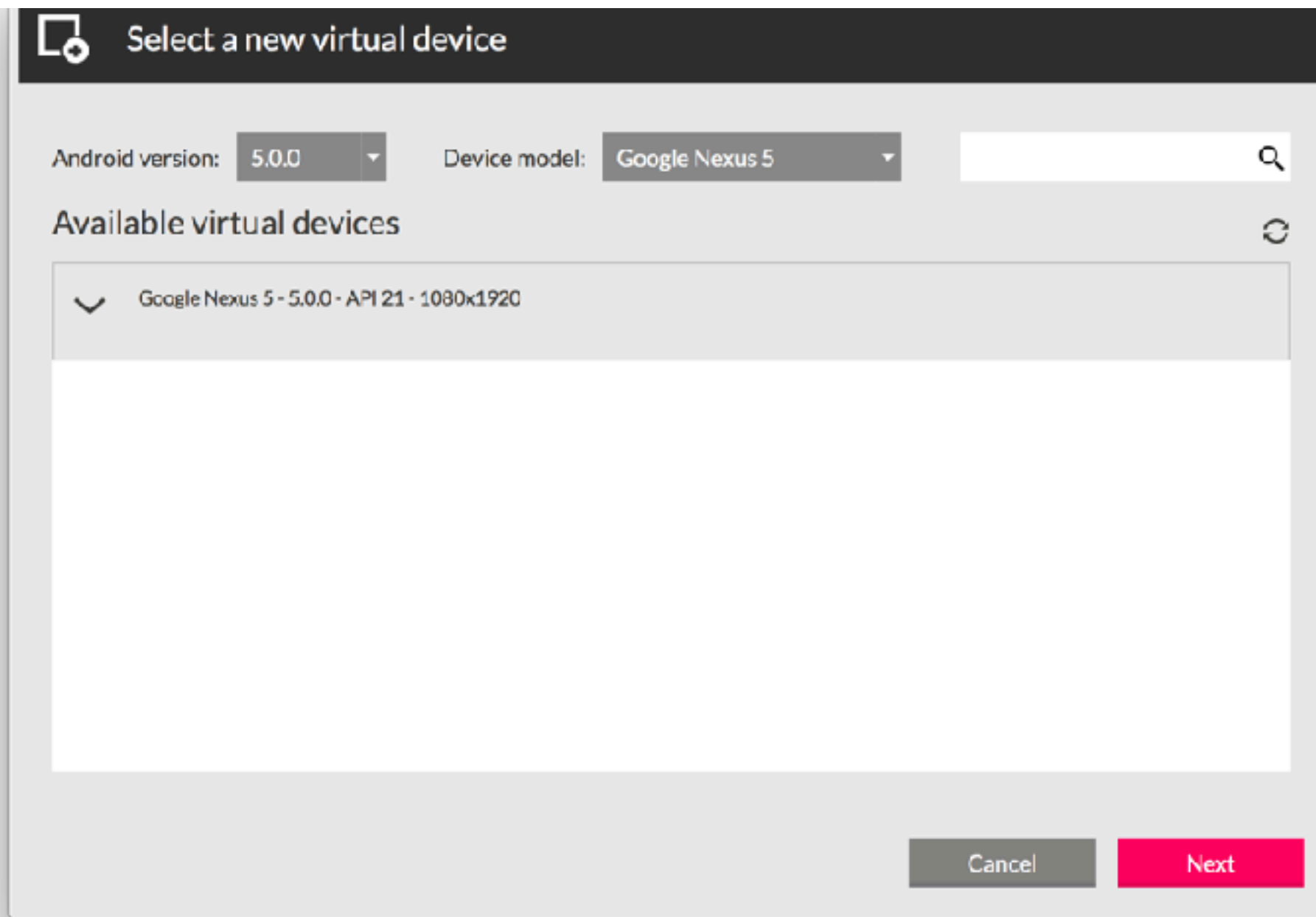
กรอกข้อมูล

ตามที่ได้ลงทะเบียนไว้ที่ web genymotion



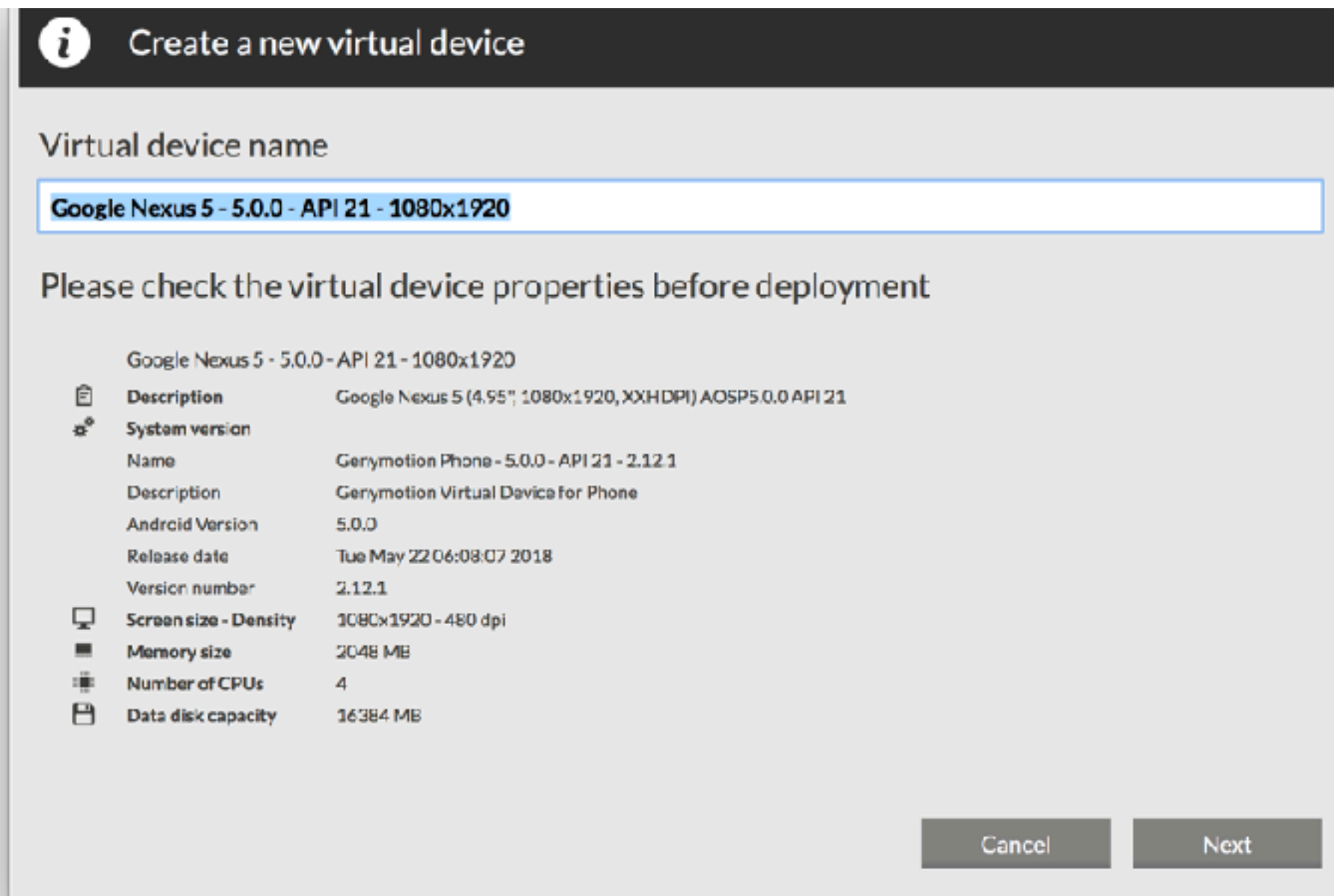
กดปุ่ม Add

เพื่อเข้าสู่การติดตั้ง virtual device



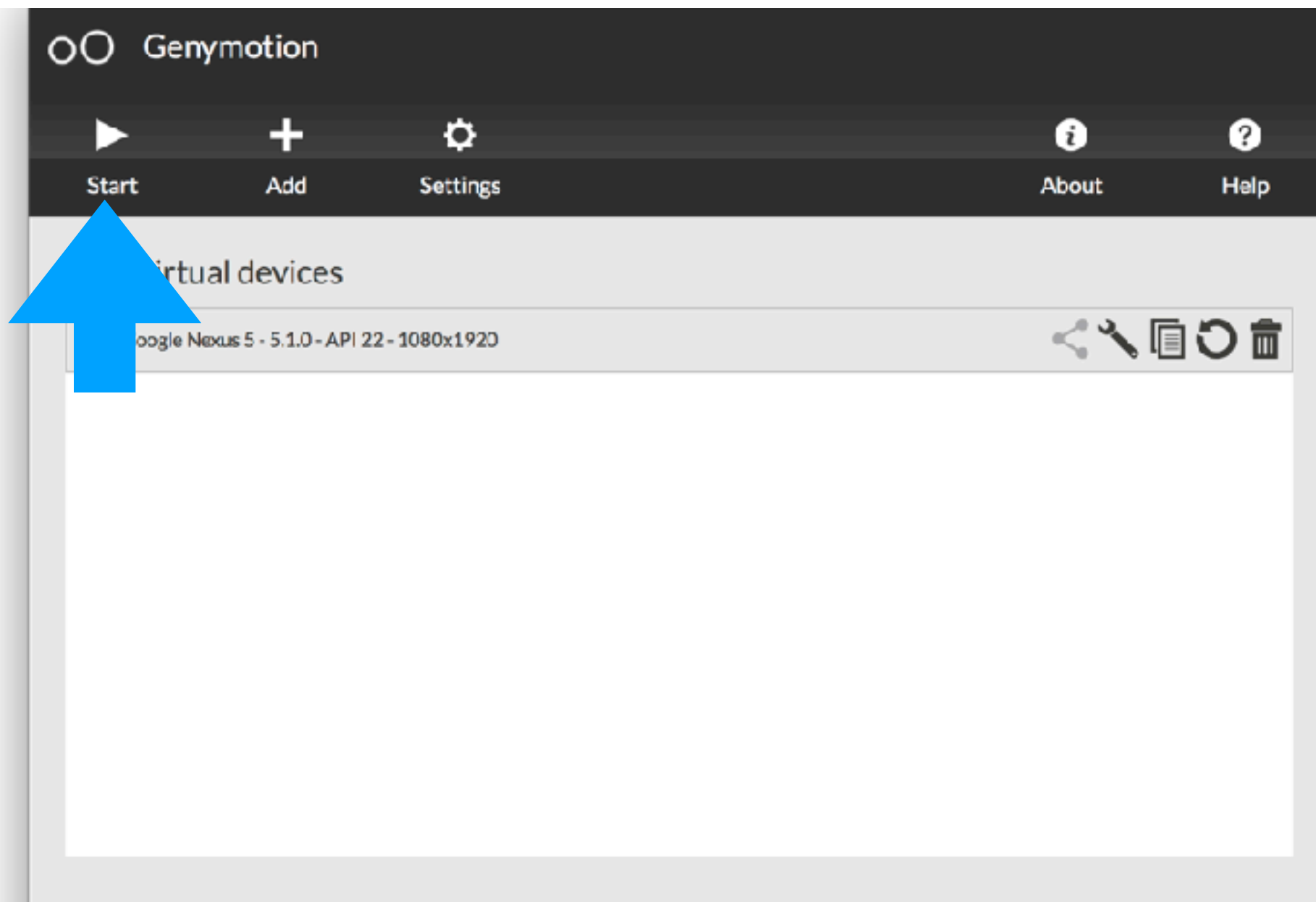
ค้นหา virtual device

เลือกตัวที่ต้องการติดตั้งแล้วกด next



ตั้งชื่อ Virtual device name

เสร็จแล้วให้กด Next



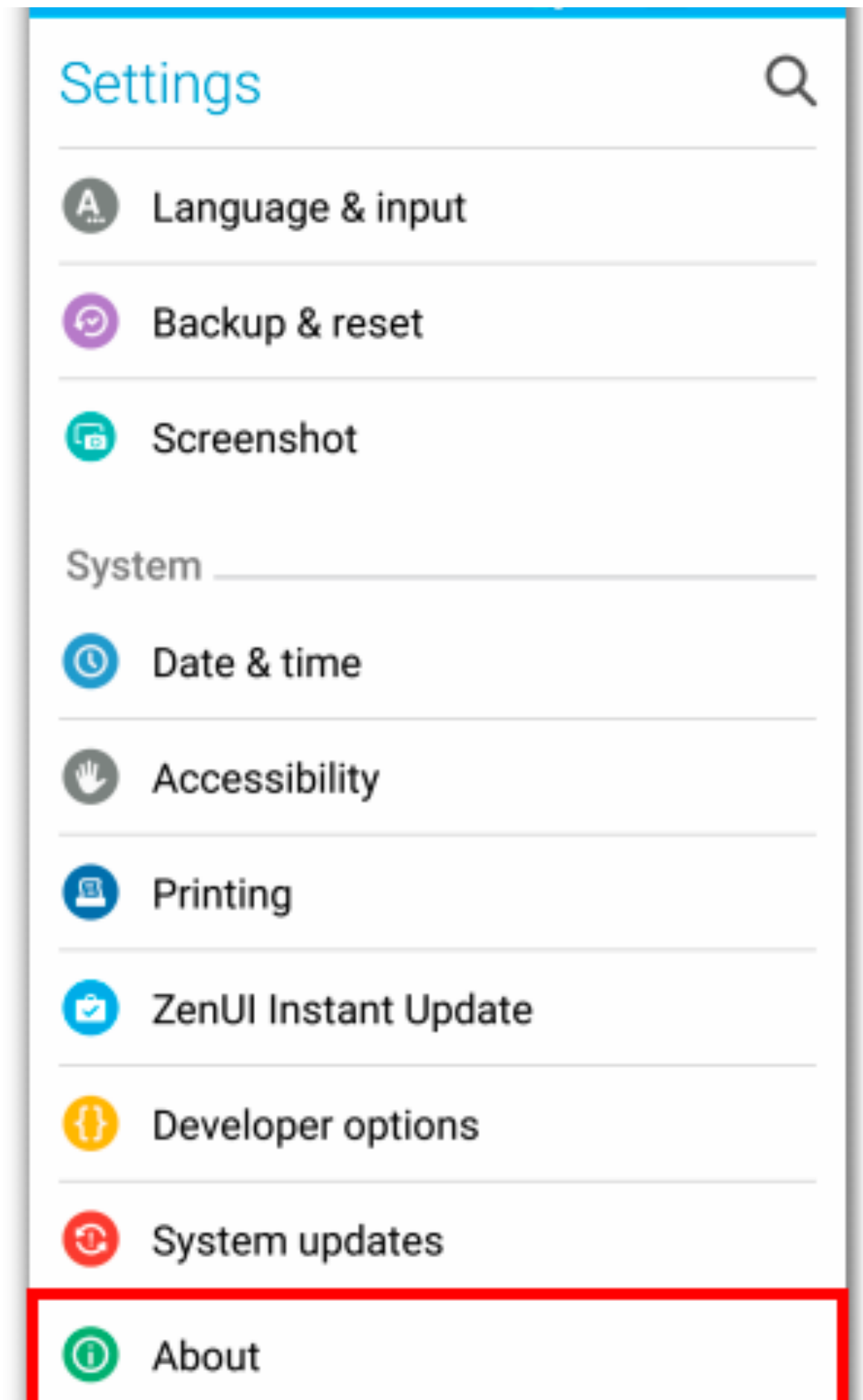
สามารถสั่ง run ได้

ด้วยการเลือก Virtual devices แล้วกด Start

วิธีการตั้งค่า Developer mode ใน Android

เลือก

Setting / About



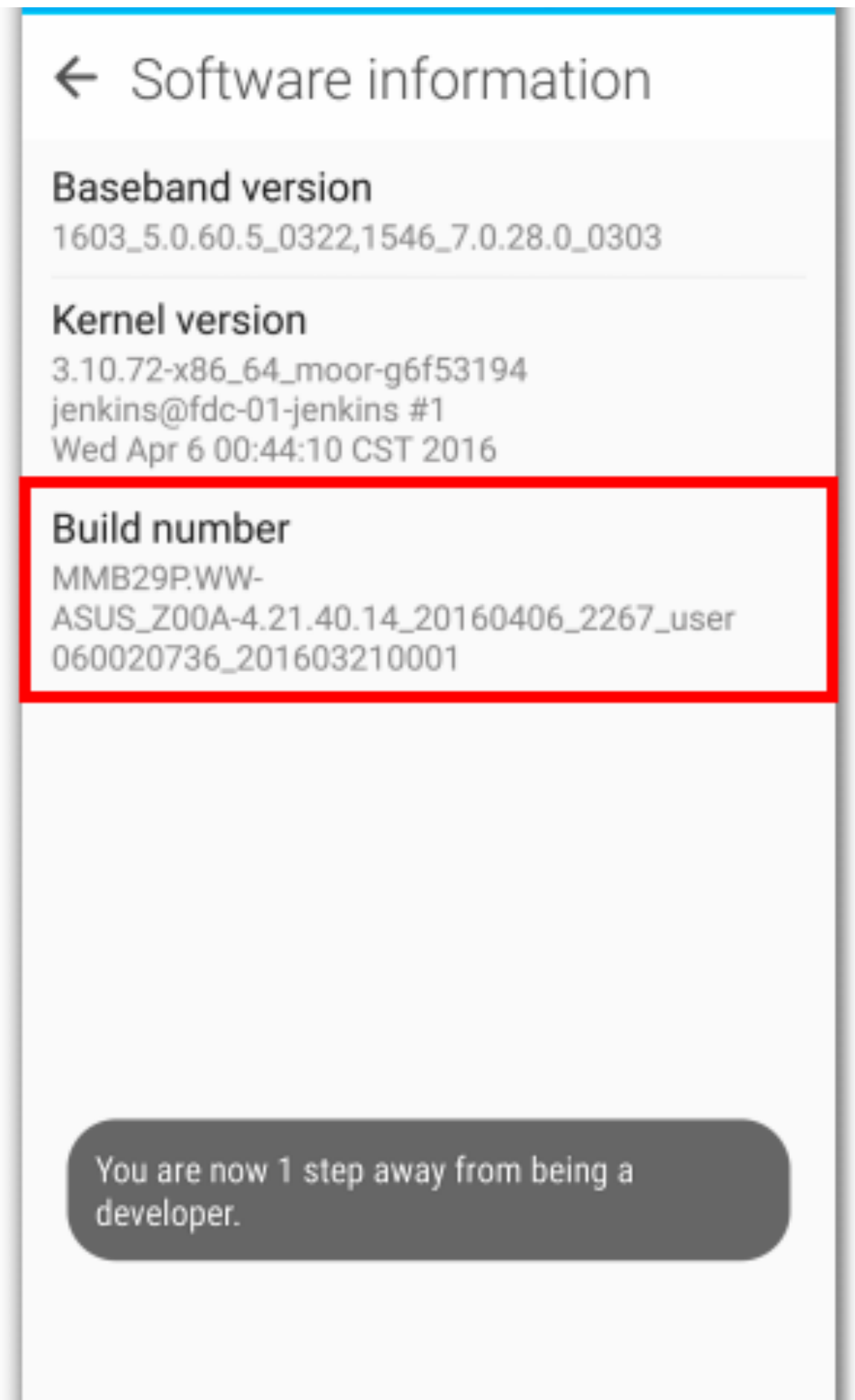
เลือก

Software information



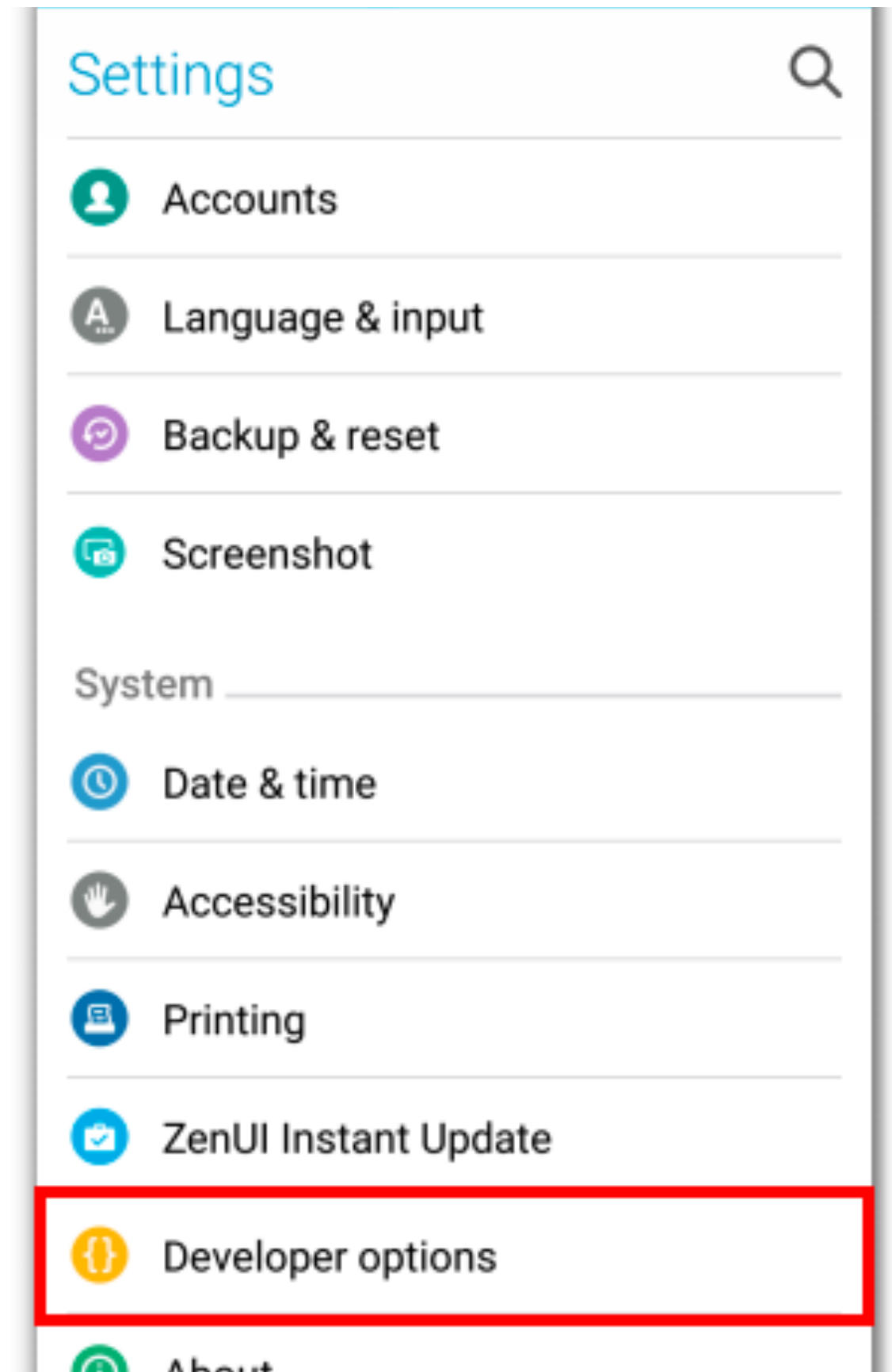
ทำดังนี้

เคาะที่ Build number 7 ครั้งเพื่อ
เปิดการทำงานของ developer
mode



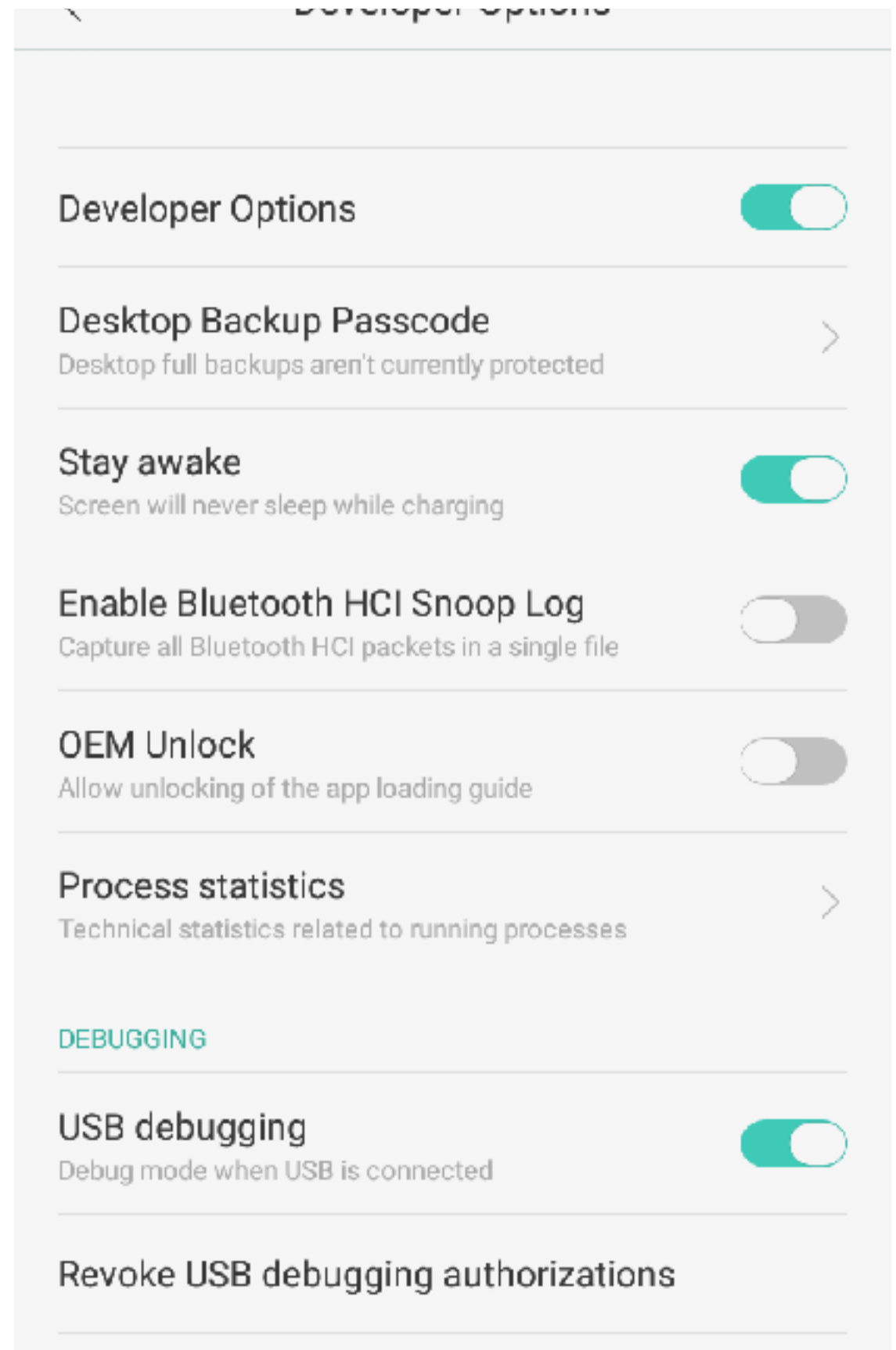
เลือก

Developer options
เพื่อเข้าไปตั้งค่าการใช้งาน



ตั้งค่า

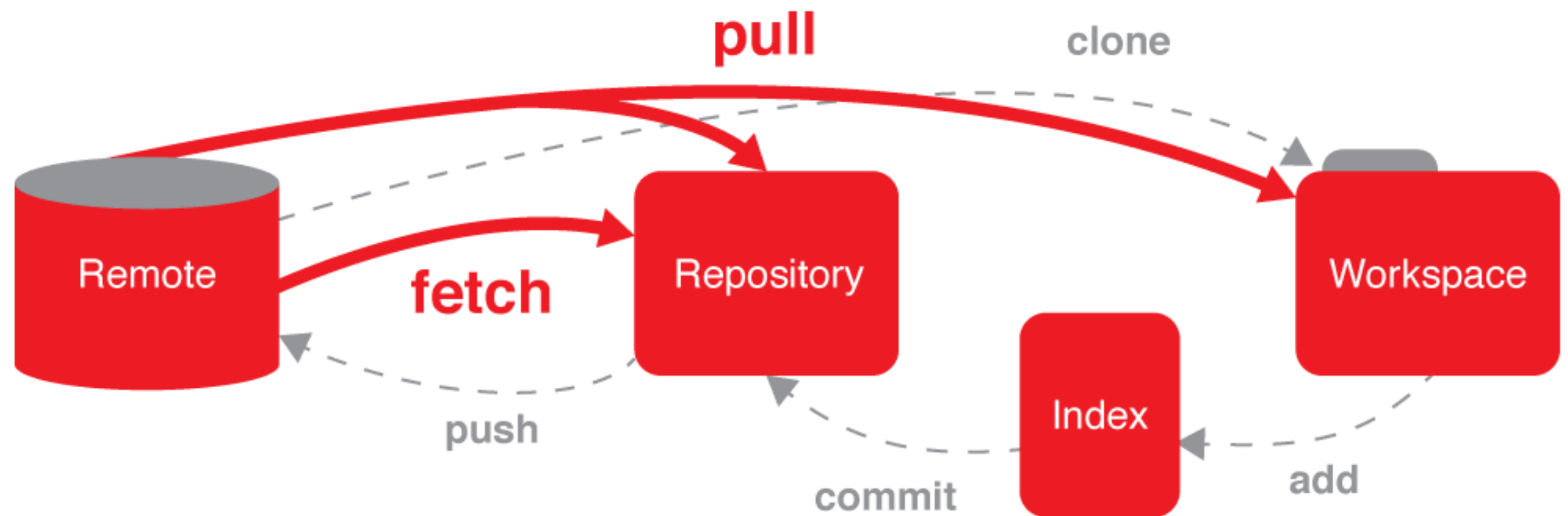
เปิดการใช้งานดังนี้
Developer Options
Stay awake
USB debugging



Version Control

With Github

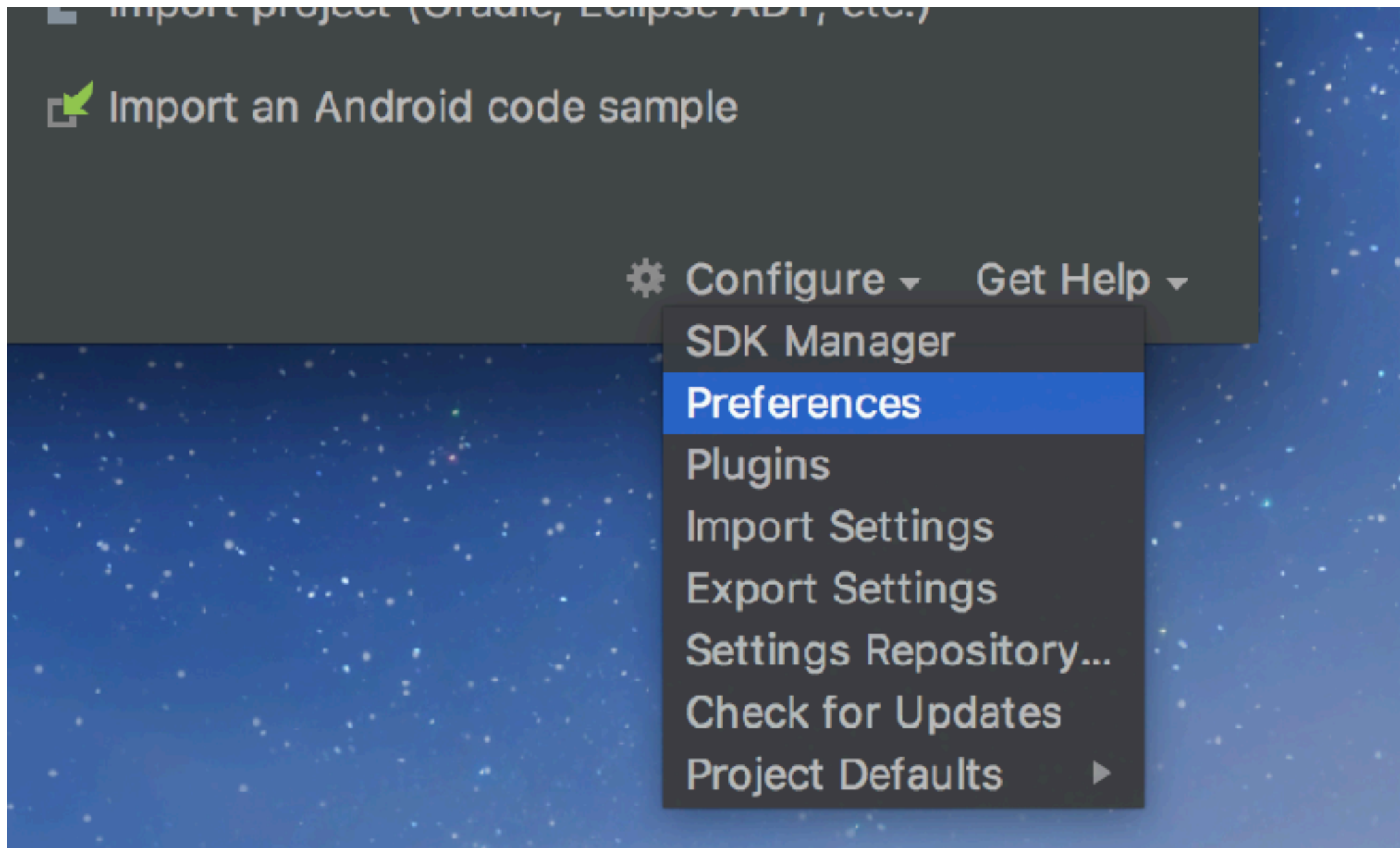
Overview



ขั้นตอน

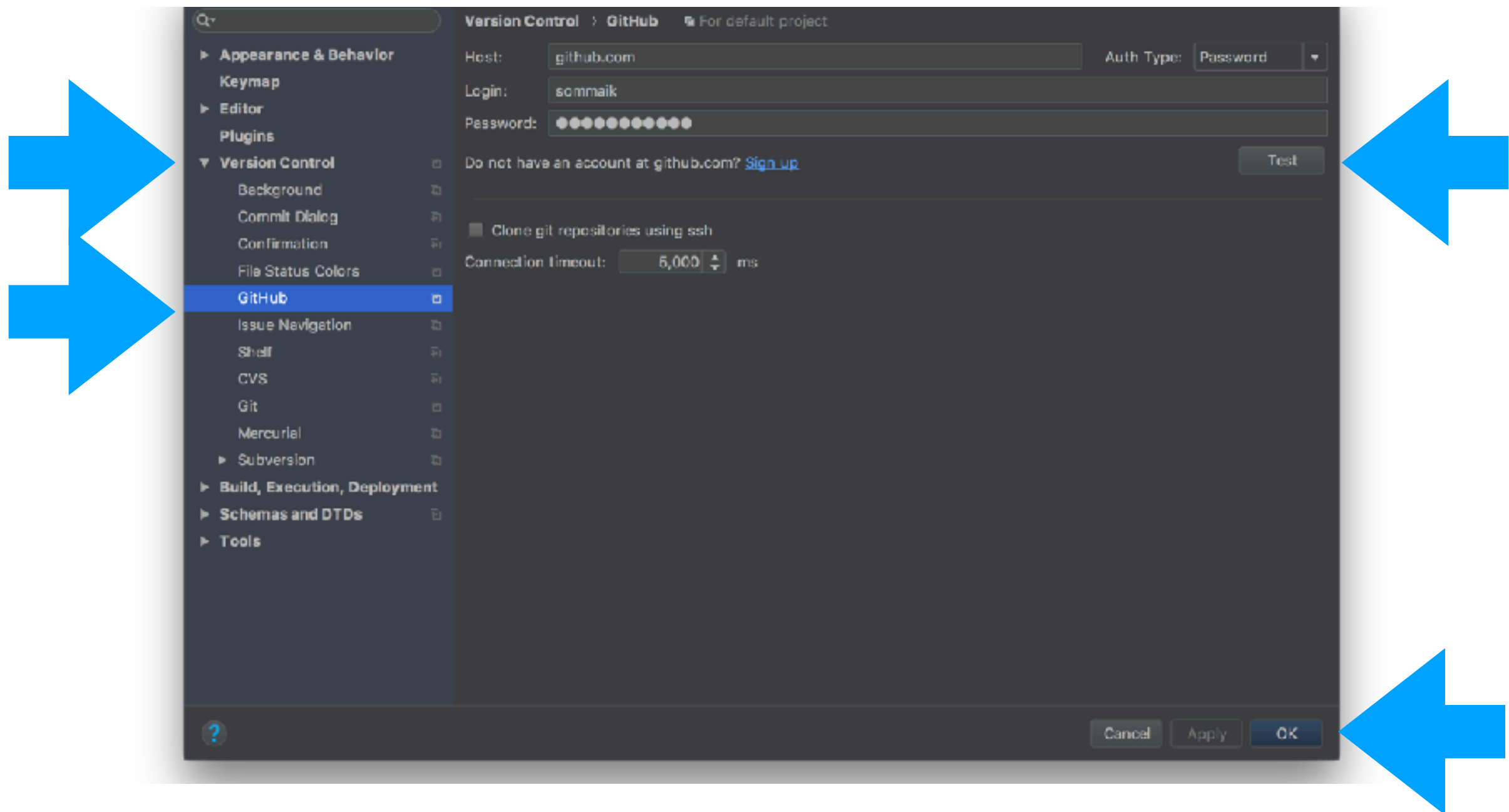
- ติดตั้งโปรแกรม git client ลงในเครื่อง
- ลงทะเบียนที่ website github.com
- ตั้งค่าผูก account กับ android studio
- Share project ไปที่ [GitHub.com](https://github.com)

ตั้งค่าผูก account กับ android
studio



กดปุ่ม Configure

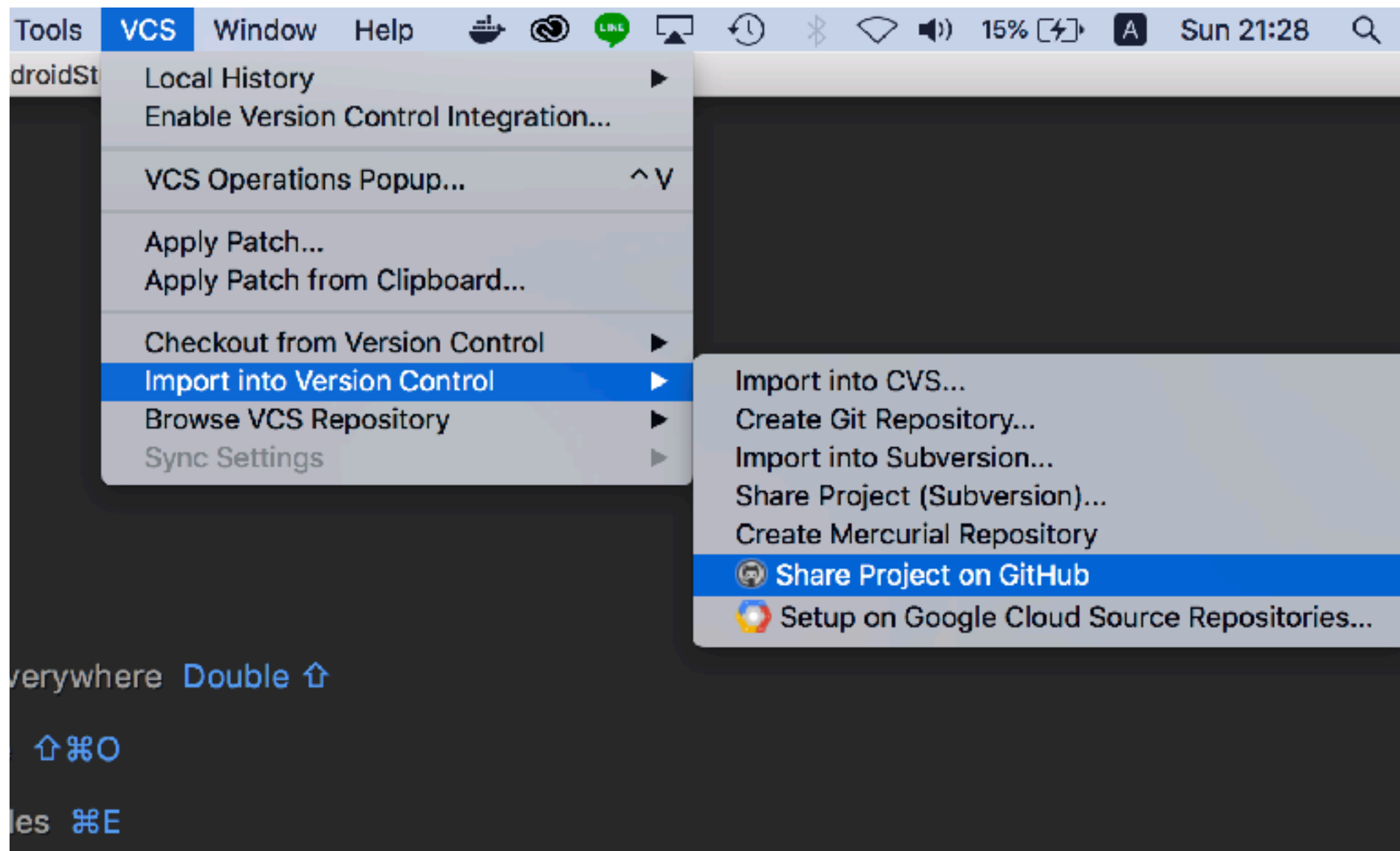
เลือก Preferences



เลือก Version Control / Github

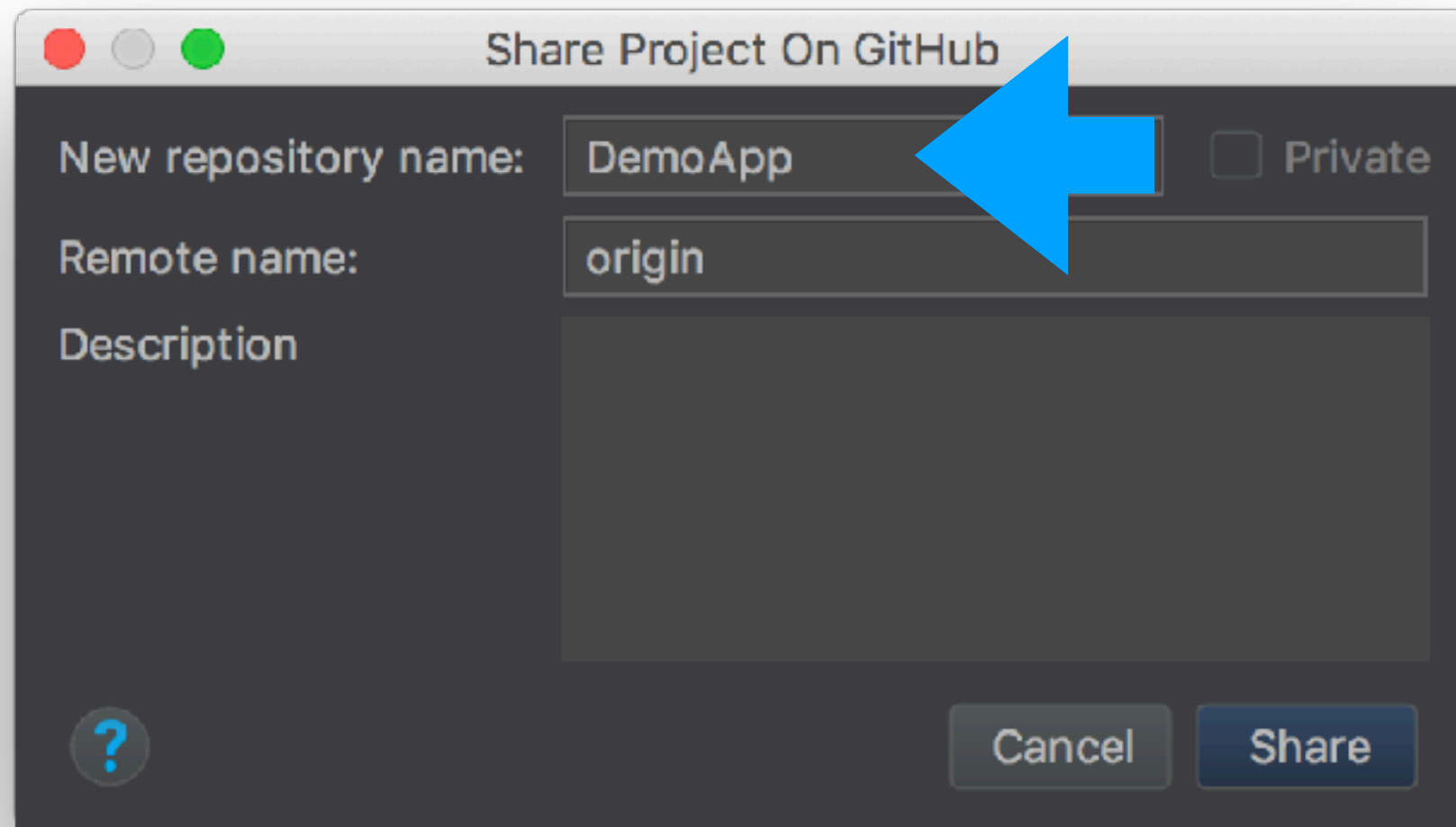
ใส่ login, password กด test ถ้าสำเร็จให้กด OK ด้านล่าง

Share project to
github.com




ກາງ VCS / Import into Version Control

ເລືອກ Share Project on GitHub



ใส่ชื่อ name

เสร็จแล้วให้กด Share

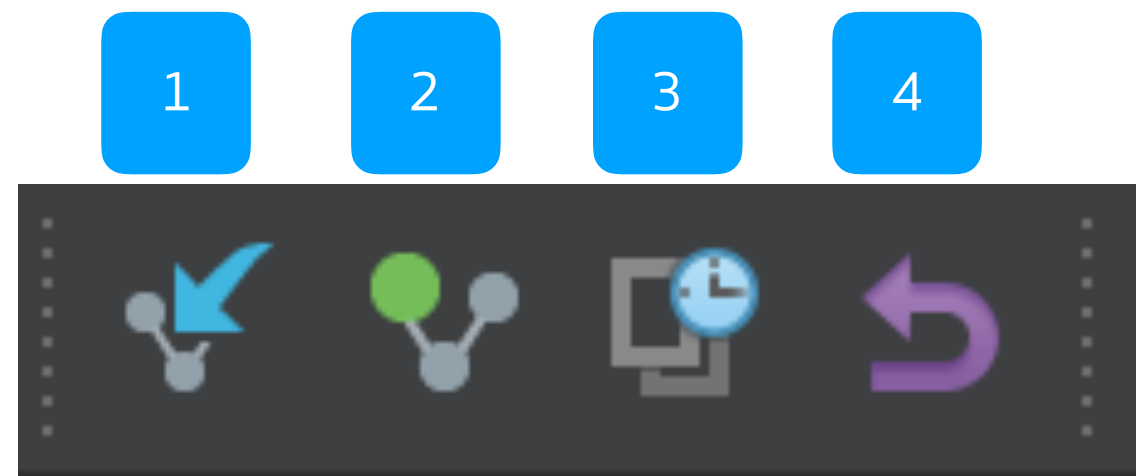
 Sommaik finish 1August		Latest commit e
📁 .idea	finish 1August	
📁 app	finish 1August	
📁 gradle/wrapper	Initial commit	
📄 .gitignore	Initial commit	
📄 build.gradle	Initial commit	
📄 gradle.properties	Initial commit	
📄 gradlew	Initial commit	
📄 gradlew.bat	Initial commit	
📄 settings.gradle	Initial commit	
Help people interested in this repository understand your project by adding a README.		

ตรวจสอบการ share code

เข้า url https://github.com/your_acctount/your_project

เครื่องมือสำหรับ Git

- 1 สำหรับดึง code ล่าสุดจาก server ลงมาที่เครื่อง
- 2 สำหรับ sync code ขึ้นไปบน server
- 3 ดูประวัติเก่า
- 4 ย้อนกลับ



วิธีการ commit code

