

คู่มือประกอบการเรียน

Flutter Crash Course

เขียน Dart และ Flutter อย่างเข้าใจและประยุกต์ใช้งานได้

อนุชิต ชโลธ

สำนักพิมพ์ท้อปวง

Flutter Crash Course

เขียน Dart และ Flutter อย่างเข้าใจและประยุกต์ใช้งานได้

พิมพ์ครั้งที่ 3

อนุชิต ชโลธร

สำนักพิมพ์ก๊อปวาง

คำนำ

หนังสือเล่มนี้เป็นสรุปเอกสารประกอบการฝึกอบรม Flutter Crash Course Season 1-2 โดยเนื้อหาอัปเดตจากการฝึกอบรมในช่วงปี 2022 ที่ผ่านมา อัปเดตเนื้อหาให้ครอบคลุม Flutter เวอร์ชันใหม่, Material Design 3 และ FlutterFire เวอร์ชันใหม่

วัตถุประสงค์ของหลักสูตร

1. เข้าใจการออกแบบหน้าจอ การเลือกใช้ Widget
2. เข้าใจการใช้งานภาษา Dart เบื้องต้น
3. เข้าใจการใช้ State Management
4. เข้าใจการเชื่อมต่อกับ APIs
5. เข้าใจการเชื่อมต่อกับ Firebase

ตัวอย่างที่ใช้ในการฝึกอบรม จะแบ่งเป็น 7 ตัวอย่าง ดังนี้

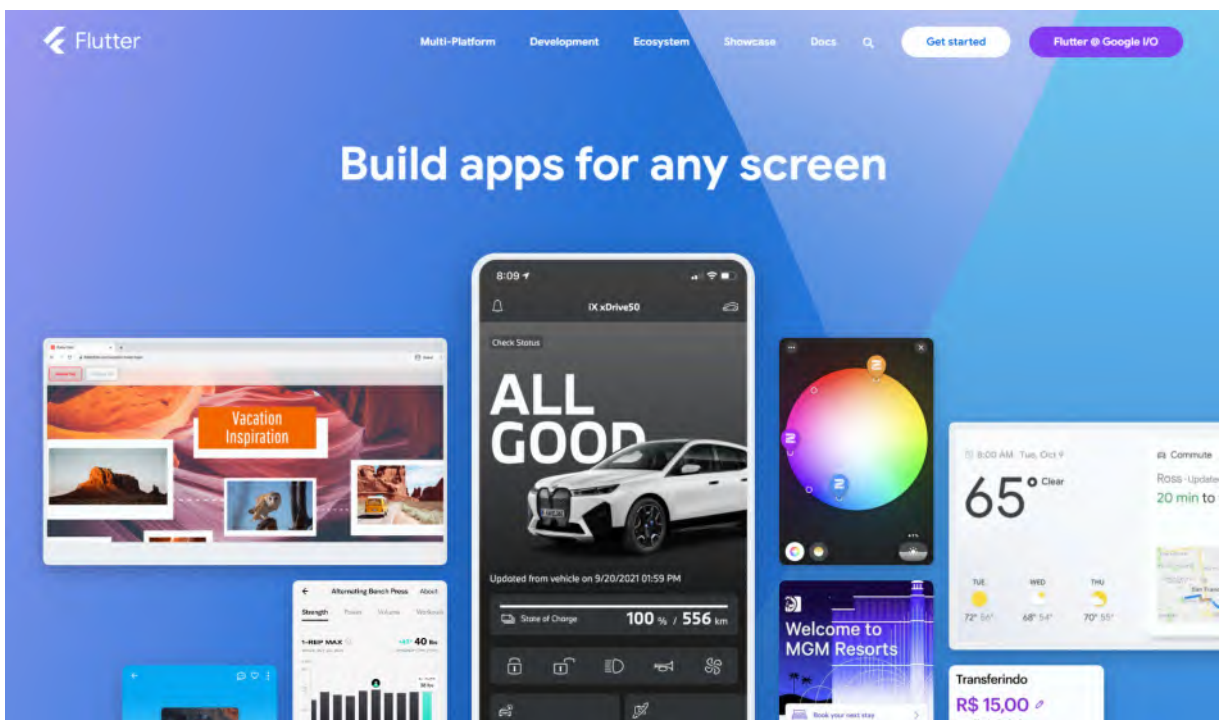
1. นามบัตร (Business Card)
2. ปั้นแปะ หัวท้าย (Head and Tail)
3. คำนวณดัชนีมวลกาย (BMI)
4. คำถาม (Quiz)
5. รายงานพยากรณ์อากาศ (Weather Forecast)
6. รายการที่ต้องทำ (Todo List)
7. แชท (Chat)

สารบัญ

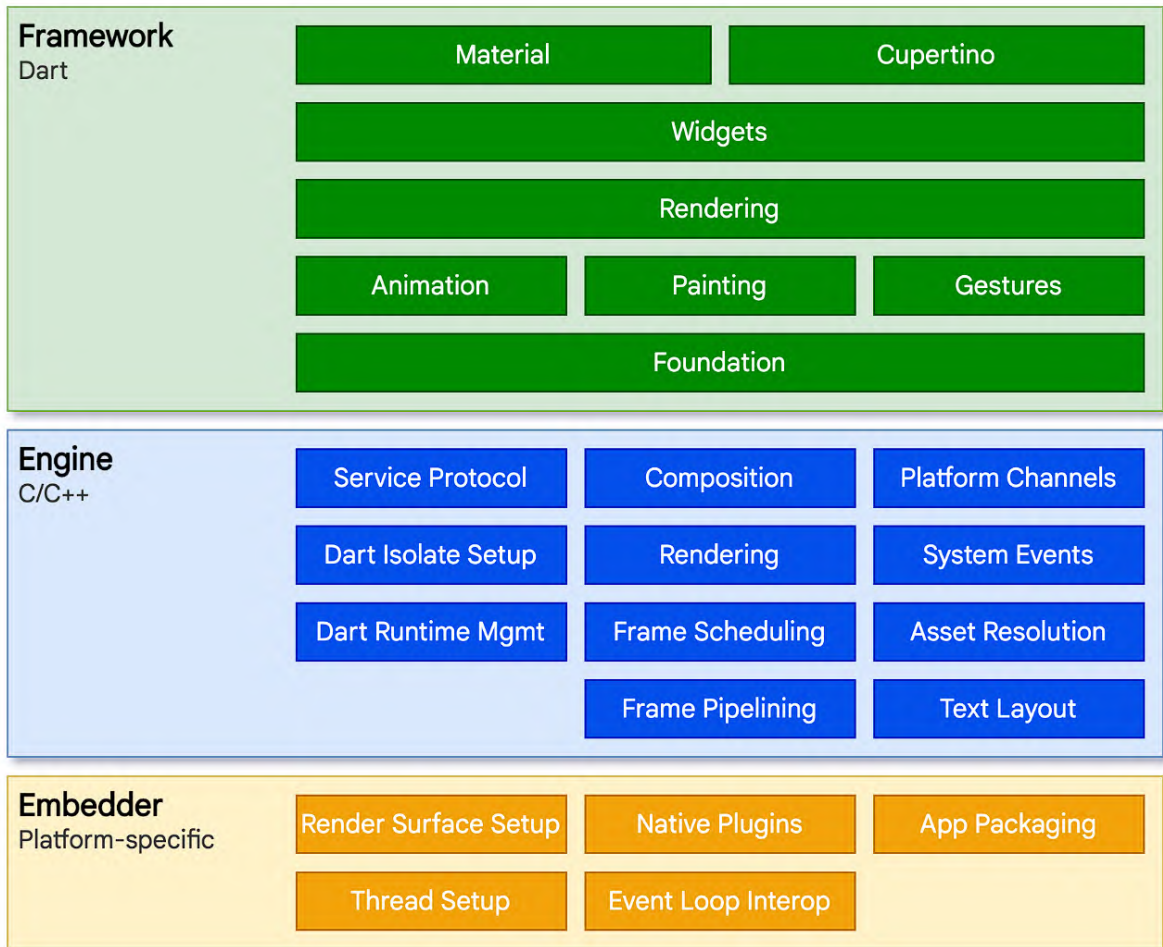
| | |
|---|-----|
| คำนำ | 1 |
| แนะนำ Flutter | 3 |
| ติดตั้ง Flutter และเครื่องมือต่างๆ | 7 |
| ตัวอย่าง นามบัตร (Business Card) | 34 |
| ตัวอย่าง ปิ่นแปะ หัวก้อย (Head & Tail) | 63 |
| ตัวอย่าง คำนวณดัชนีมวลกาย (BMI) | 73 |
| ตัวอย่าง คำถาม (Quiz) | 88 |
| ตัวอย่าง พยากรณ์อากาศ (Weather Forecast) | 101 |
| แนะนำ Firebase | 117 |
| ตัวอย่าง รายการที่ต้องทำ (Todo List) | 125 |
| ตัวอย่าง แอปแชท (Chat) | 154 |
| ดาวნიโหลดซอร์สโค้ด | 171 |
| แนะนำหนังสือชุด “ก๊อปปวาง” | 174 |
| สูตรลัด Flutter | 174 |
| สูตรลัด GetX | 175 |
| สูตรลัด FlutterFire | 176 |
| สูตรลัด Hasura | 177 |
| คู่มือประกอบการเรียน Flutter Crash Course | 178 |
| Material Design 3 | 179 |
| สูตรลัด Dart 3 | 180 |
| สูตรลัด Dart Frog | 182 |
| สูตรลัด Appwrite | 183 |
| สูตรลัด Serverpod | 185 |
| Prisma Client Dart | 186 |

แนะนำ Flutter

หากคุณเคยได้ยินว่า เขียนแอปด้วย Flutter สามารถสร้างแอป บนมือถือ แท็บเล็ต ทั้ง Android, iOS รวมไปถึงแอปบน macOS, Windows, Linux และ Embedded System ต่างๆ ได้ ฟังแล้วดูเหมือนเป็นเครื่องมือวิเศษ แต่อย่าเพิ่งหลงเชื่อจนเกินไป จนกว่าคุณจะได้เข้าใจว่า Flutter คืออะไรกันแน่

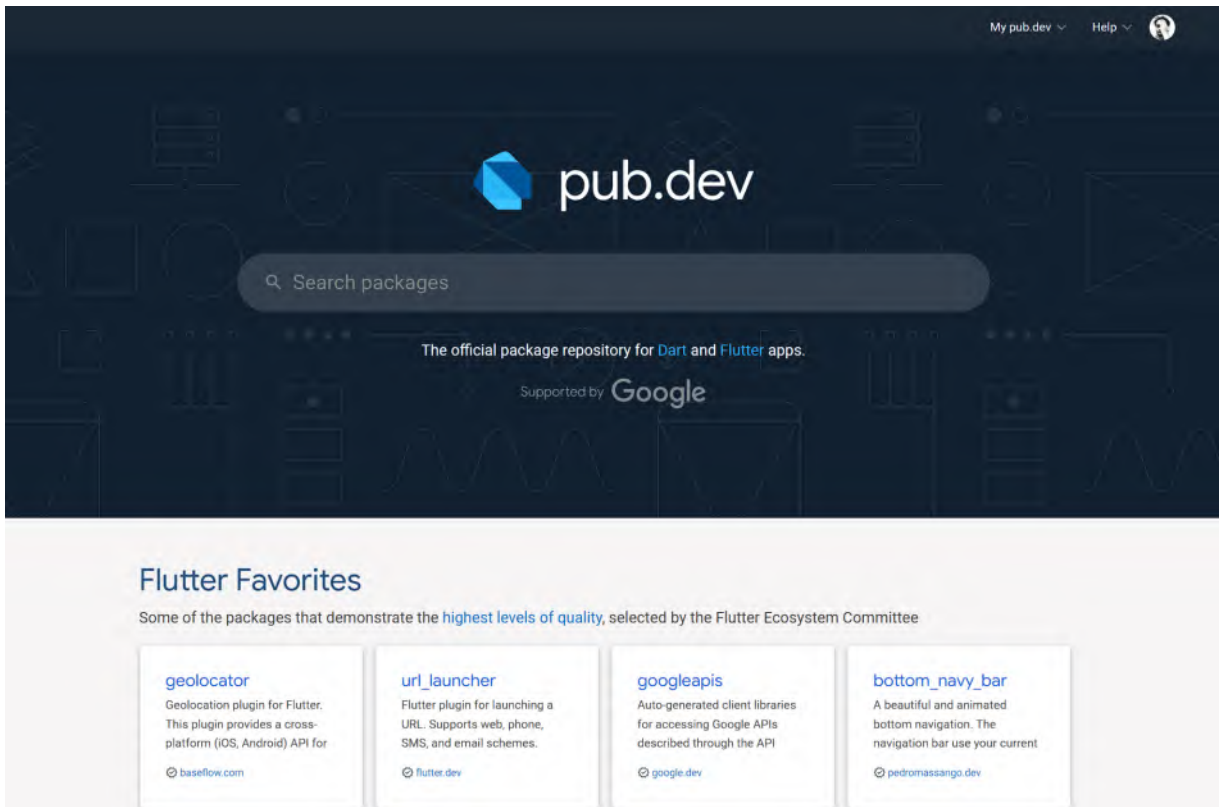


Flutter เป็น UI Toolkits ที่สามารถสร้างแอปสวยๆ ได้อย่างง่ายดาย และที่สำคัญคุณต้องเขียนภาษา Dart (ไม่ใช่เขียนภาษา Flutter นะ) ซึ่ง Dart นี้แหละจะเป็นเครื่องมือที่ช่วยให้คุณสร้างแอปไปใช้งานในแพลตฟอร์มต่างๆ ได้

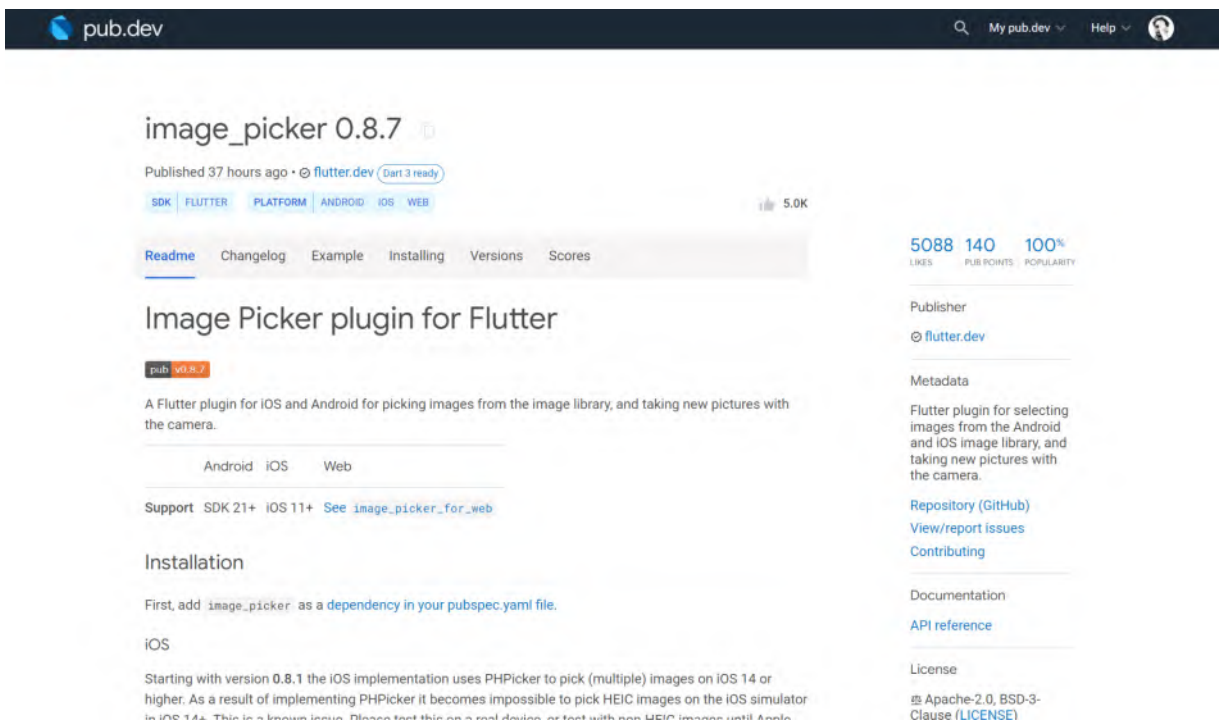


จากภาพจะเห็นว่าส่วน UI ต่างๆ จะอยู่ข้างบน เป็นส่วนเชื่อมต่อกับผู้ใช้ ซึ่งก็คือ Flutter หรือ UI Toolkits นั้นเอง ในส่วนประสานงาน การแสดงผลกราฟิก แอนิเมชันสวยๆ จะใช้ render engine อย่าง Skia (ในอนาคตจะเปลี่ยนเป็น Impeller) เข้ามาช่วย Skia สามารถเรนเดอร์หน้าจอได้ถึง 60fps เลยทีเดียว

ในส่วนการทำงานเชื่อมต่อกับแพลตฟอร์มต่างๆ ยังต้องใช้ Dart ในการเชื่อมต่อผ่าน Plugin ต่างๆ จึงจะสามารถเรียกใช้งาน lib ในแต่ละแพลตฟอร์มได้ ตัวอย่างเช่น File Picker, Image Picker, Location service เป็นต้น



ซึ่งจะต้องดูว่า Plugin ที่ต้องการใช้ รองรับแพลตฟอร์มที่ต้องการเอาไปใช้งานหรือไม่ ตัวอย่างเช่น Image Picker



จะเห็นว่า Image Picker สามารถใช้งานได้กับ Android, iOS และ Web เท่านั้น หากต้องการใช้กับแพลตฟอร์มอื่นๆ เช่น Windows, macOS ต้องเลือกใช้ Plugin อื่นแทน

ทำไมบริษัทเทคโนโลยีหลายบริษัทใช้ Flutter เอาไปพัฒนาผลิตภัณฑ์ของตัวเอง คำตอบคงหนีไม่พ้น คำโฆษณาเหล่านี้

1. พัฒนาง่ายและรวดเร็ว
2. พัฒนางานตามงานออกแบบได้อย่างอิสระและสวยงาม
3. การเรนเดอร์หน้าจอที่เร็ว
4. สามารถสร้างแอปรองรับแพลตฟอร์มต่างๆ ได้โดยใช้โค้ดเดียวกัน

หากคุณอยากรู้ว่าเขียนแอปโดยใช้ Flutter เป็นอย่างไรก็คงต้องลงมือเขียนถึงจะรู้ และอย่าท้อปว้าง 🙌

ติดตั้ง Flutter และเครื่องมือต่างๆ

การเขียนแอปด้วย Flutter เราจะต้องมีเครื่องมือ 3 ส่วน ได้แก่ Flutter SDK + Dart, Toolchain สำหรับแพลตฟอร์มที่ต้องการพัฒนา และ IDE สาเหตุที่ต้องอธิบายเรื่องนี้ก่อนที่จะติดตั้งเครื่องมือต่างๆ เนื่องจากคำโฆษณาในบทที่แล้ว ซึ่งหลายคนเข้าใจว่าใช้ Flutter แล้วสามารถ build แล้วเอาไปใช้งานในแพลตฟอร์มต่างๆ ได้เลย แต่ช้าก่อน!

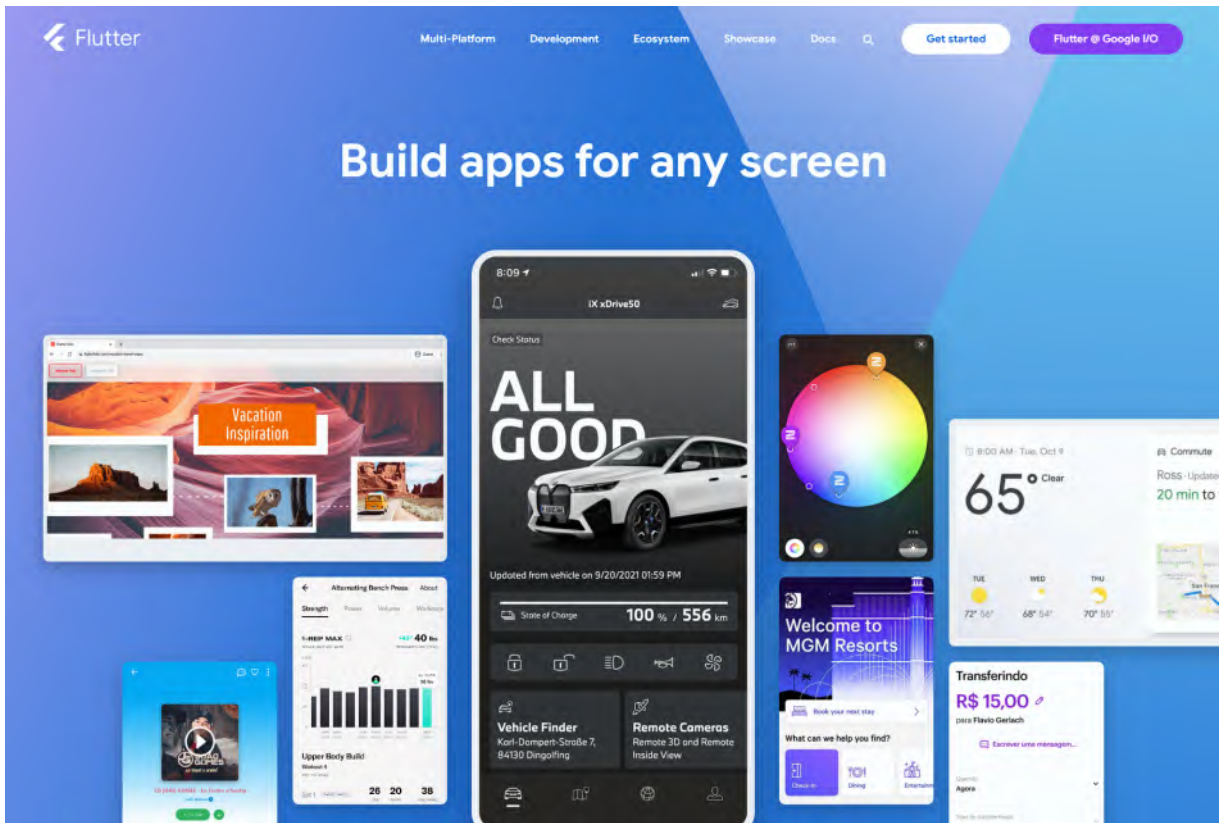
หากคุณต้องการเขียนแอปสำหรับ iOS, macOS คุณต้องใช้ Toolchain ของแพลตฟอร์ม iOS และ macOS นั่นคือ XCode และ XCode ติดตั้งได้เฉพาะ macOS เท่านั้น (เข้าใจตรงกันนะ)

หากคุณต้องการเขียนแอปสำหรับ Windows คุณต้องใช้ Toolchain ของ Windows นั่นก็คือ C++ SDK คุณจะต้องใช้เครื่องที่ติดตั้งระบบปฏิบัติการ Windows เท่านั้น (เข้าใจตรงกันนะ)

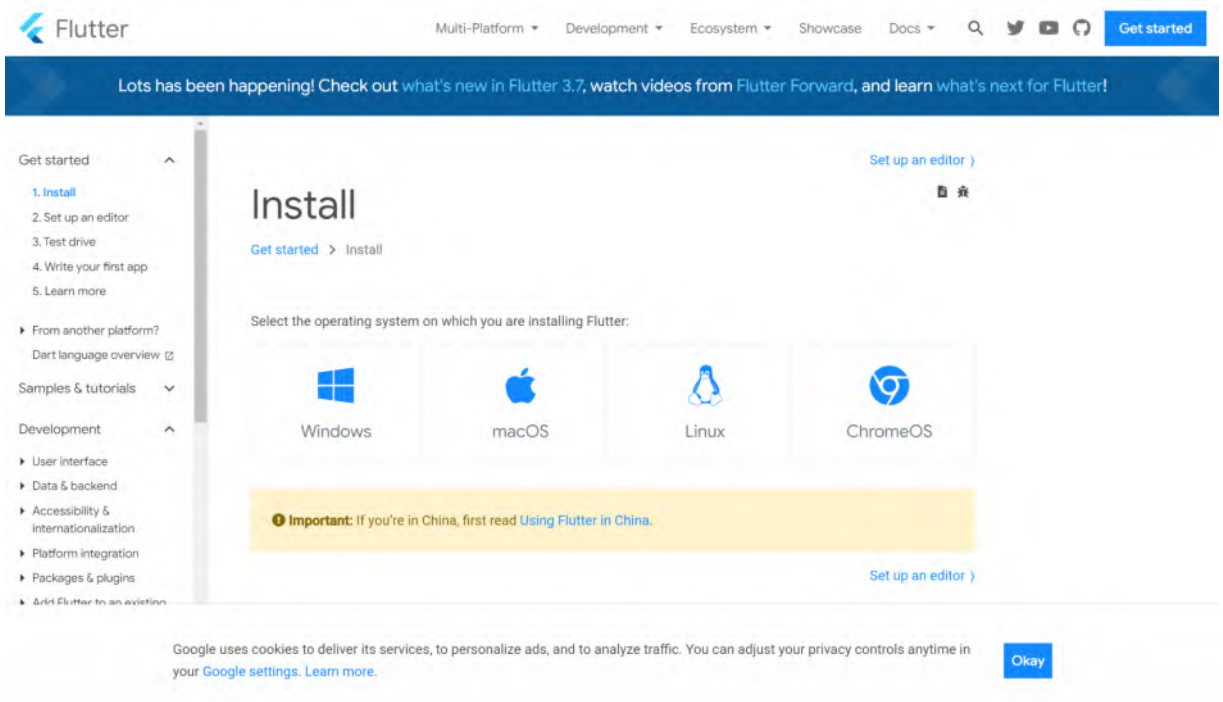
หากคุณต้องการเขียนแอปสำหรับ Linux คุณต้องใช้ Toolchain ของ Linux เช่น GTK, CMake เป็นต้น ซึ่งคุณจะต้องใช้เครื่องที่ติดตั้งระบบปฏิบัติการ Linux เท่านั้น (เข้าใจตรงกันนะ)

หากคุณต้องการเขียนแอปสำหรับ Android คุณต้องใช้ Toolchain ของ Android คือ Android SDK ซึ่งติดตั้งได้ทุกระบบปฏิบัติการ คุณสามารถใช้เครื่อง Mac, Windows, Linux, ChromeOS ก็สามารถสร้างแอปบน Android ได้ (เข้าใจตรงกันนะ)

เมื่อเข้าใจแล้วก็เข้าไปที่เว็บไซต์ flutter.dev



กดไปที่ปุ่ม Get Started



ติดตั้งเครื่องมือที่ต้องใช้งาน ในหนังสือเล่มนี้แนะนำระบบปฏิบัติการ Windows และ macOS เท่านั้น

การติดตั้ง Flutter SDK บน Windows

1. ติดตั้ง Android Studio, Android SDK ติดตั้ง Android SDK Command-line Tools
2. ติดตั้ง Visual Studio Code
3. ดาวน์โหลด Flutter SDK และแตกไฟล์ไปยัง folder ที่คุณต้องการ
4. ตั้งค่า Path ใน Environment Settings
5. เปิด Windows terminal แล้วใช้คำสั่ง flutter doctor
6. หากไม่พบ error เป็นอันใช้ได้

การติดตั้ง Flutter SDK บน macOS

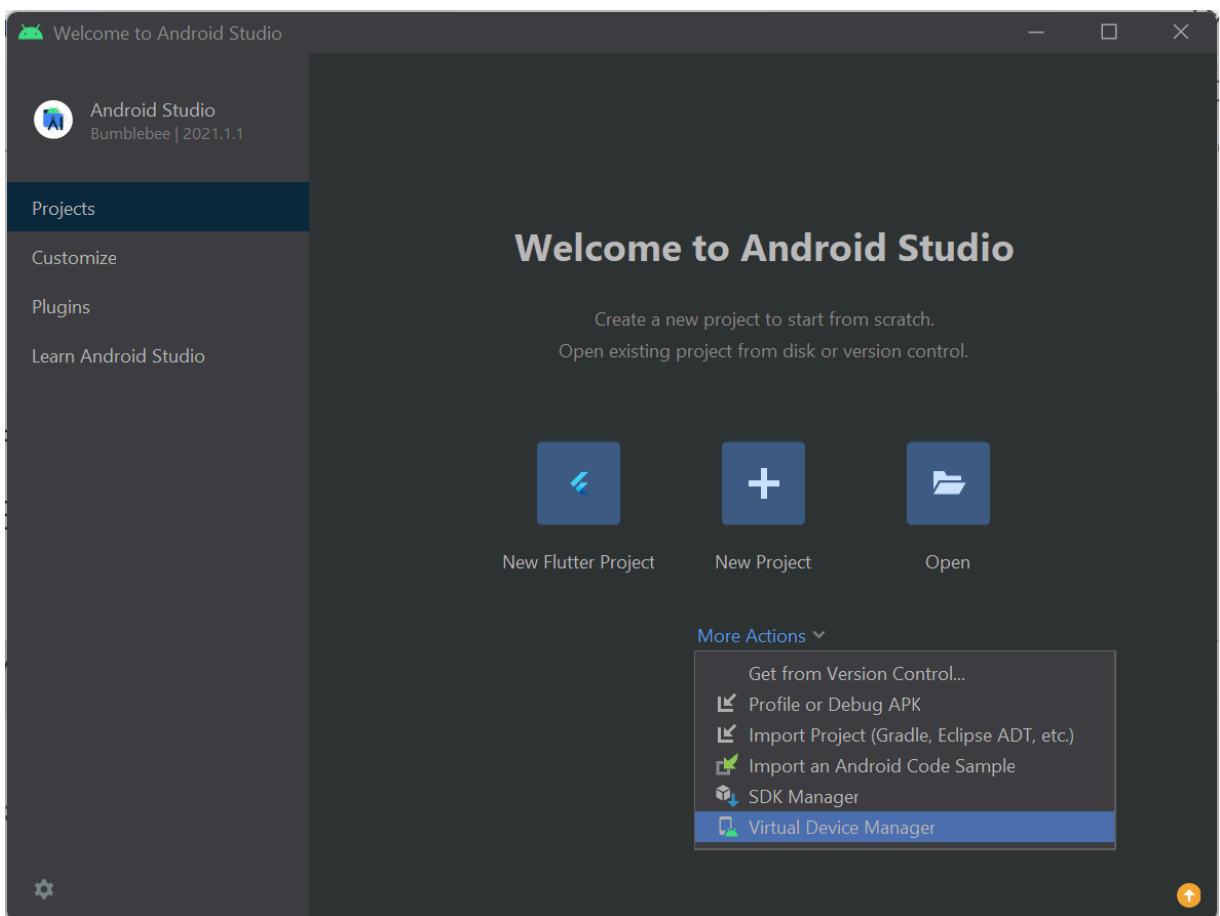
1. ติดตั้ง XCode ให้เรียบร้อย (ติดตั้งได้จาก AppStore)
2. ติดตั้ง Visual Studio Code
3. ติดตั้ง Android Studio, Android SDK ติดตั้ง Android SDK Command-line Tools
4. ดาวน์โหลด Flutter SDK และแตกไฟล์ไปยัง directory ที่คุณต้องการ ยกตัวอย่าง เช่น home directory ~/
5. ตั้งค่า Path ใน ~/.bash_profile หรือ ~/.zshrc
6. จากนั้นปิด terminal แล้วเปิดใหม่ แล้วใช้คำสั่ง flutter doctor
7. หากไม่พบ error เป็นอันใช้ได้

ก่อนจะไป Hello World ใน Flutter แนะนำให้ติดตั้ง extension ใน VSCode ดังนี้

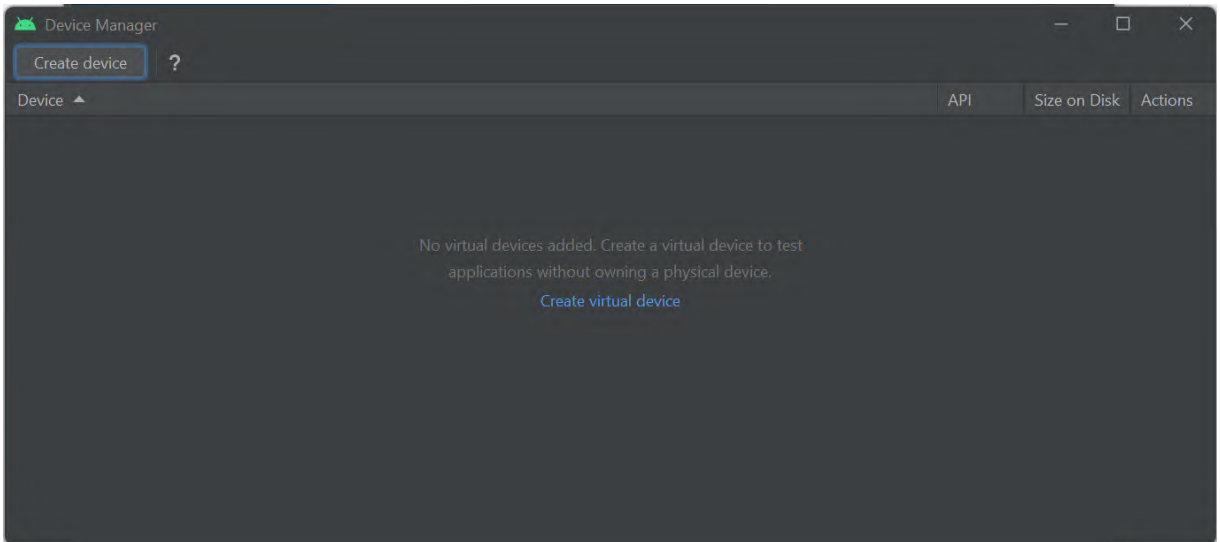
1. Awesome Flutter Snippets
2. Dart (default)
3. Dart Data Class Generator

4. Flutter (default)
5. Flutter Intl
6. Json to Dart Model
7. Todo Treex

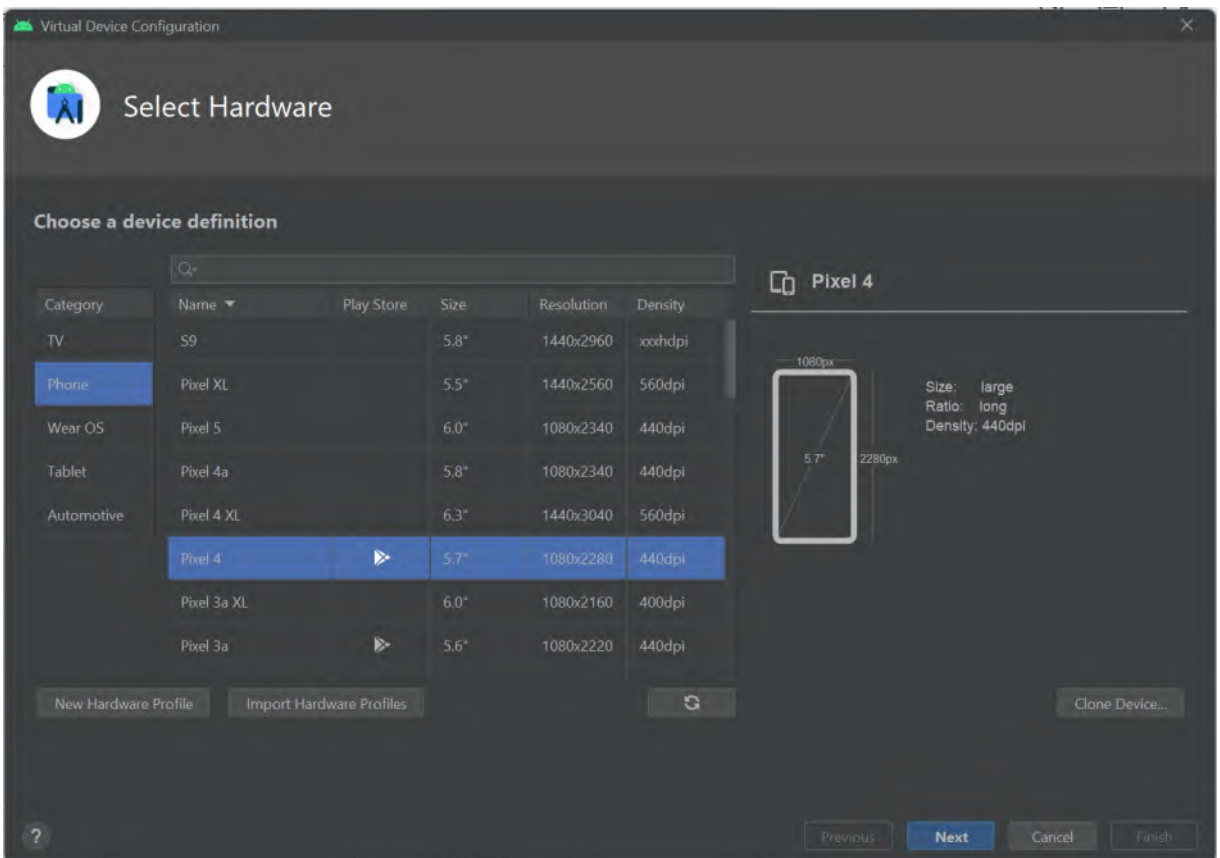
สร้าง AVD จากหน้าแรก Android Studio เลือก More Actions > Virtual Device Manager



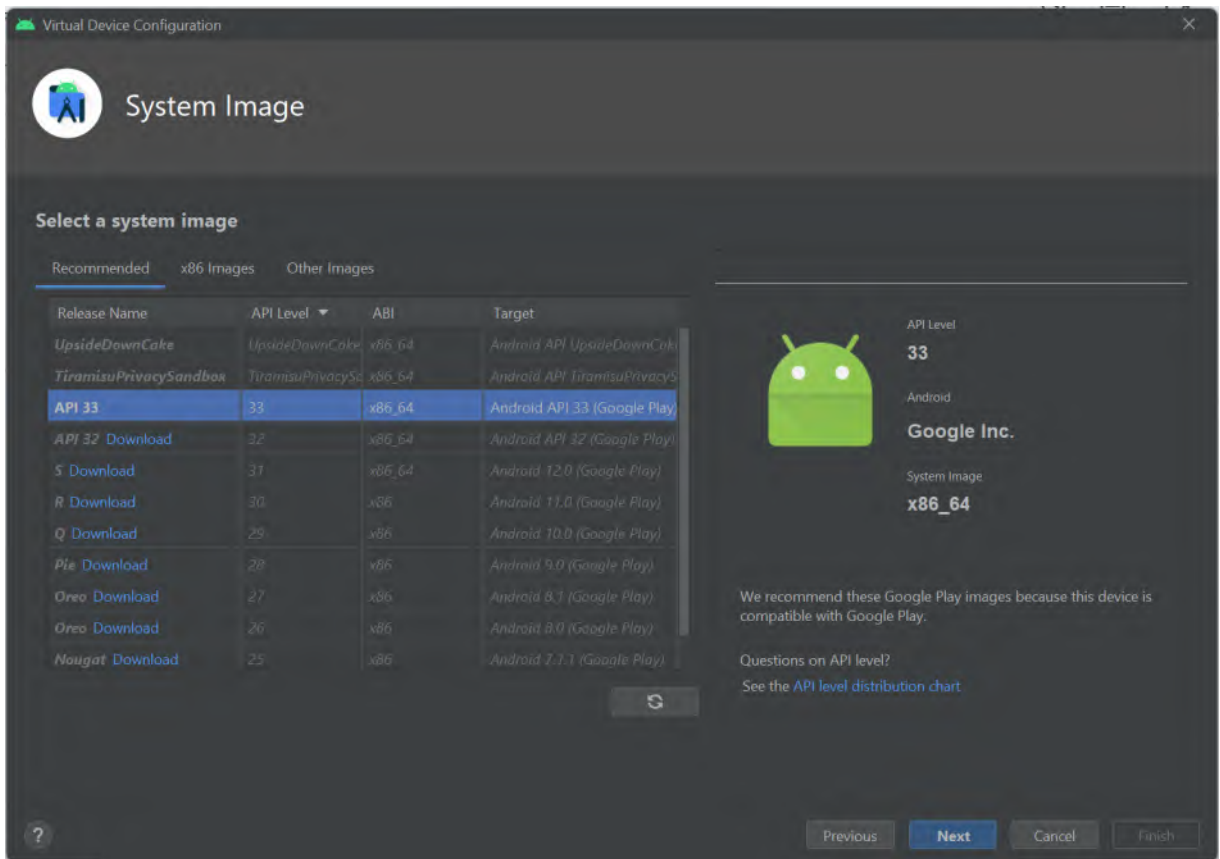
จะพบหน้าจอ Device Manager ดังภาพ



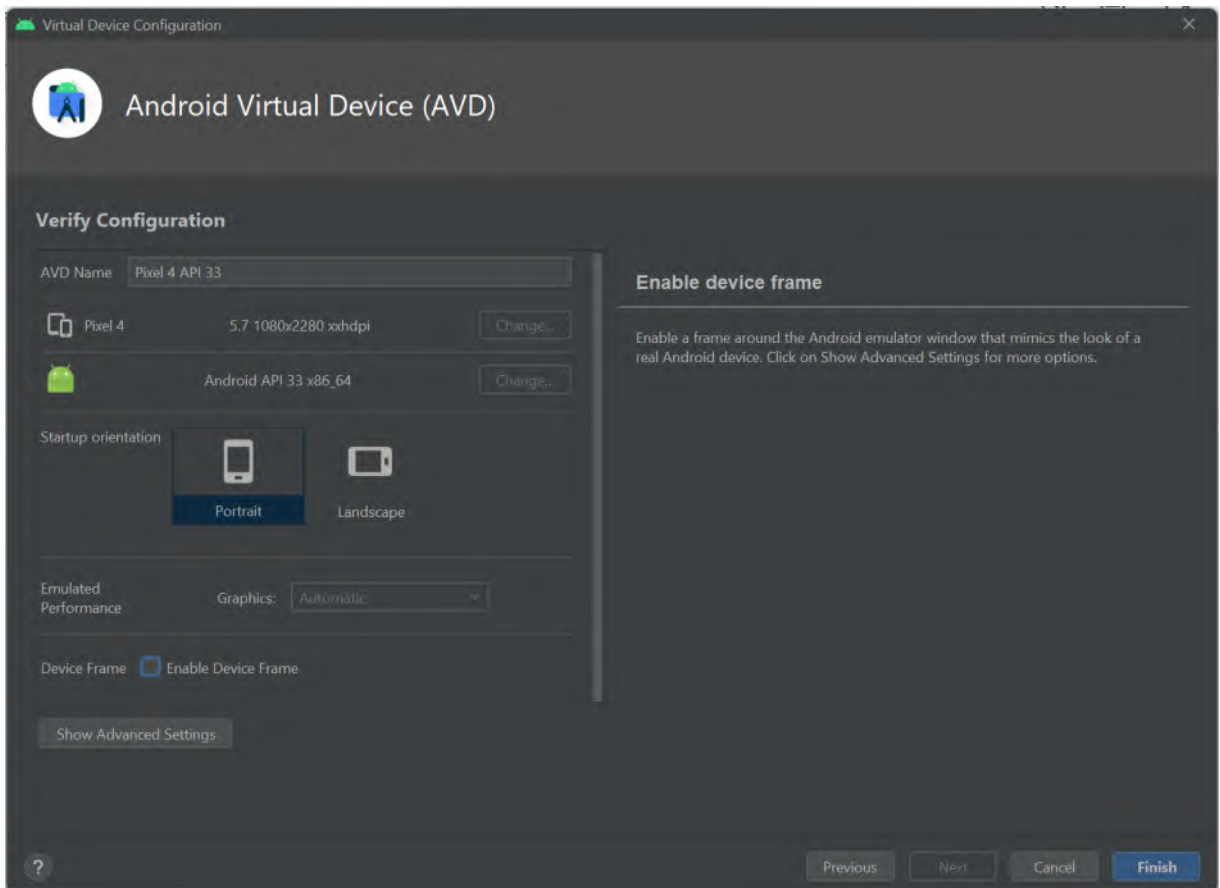
กดปุ่ม Create device เลือก Phone > Pixel 4



แนะนำให้ใช้ API 33 ขึ้นไป

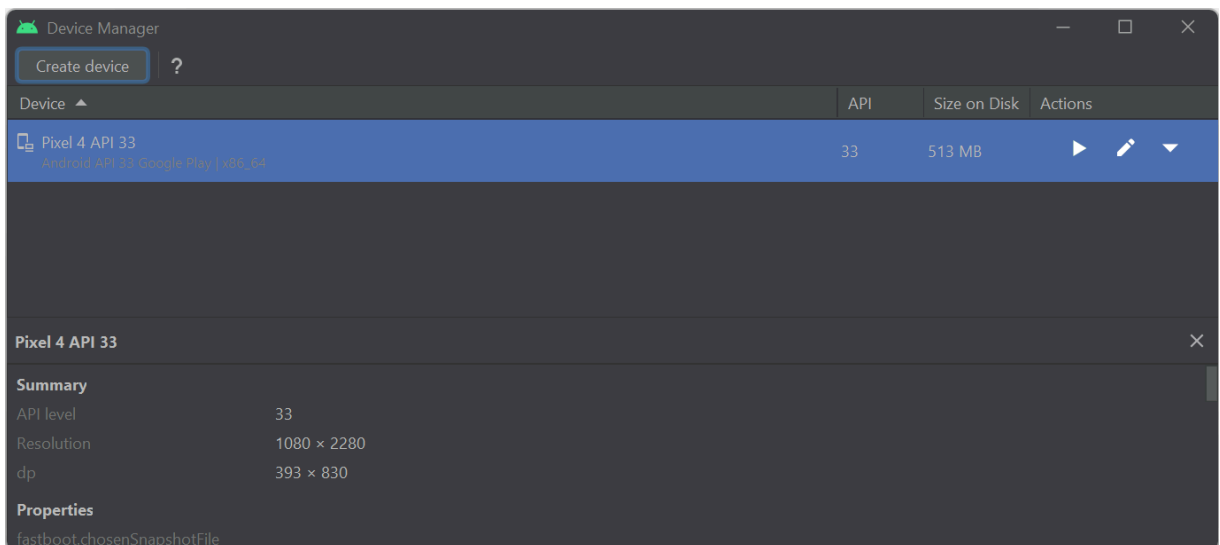


กดปุ่ม Next

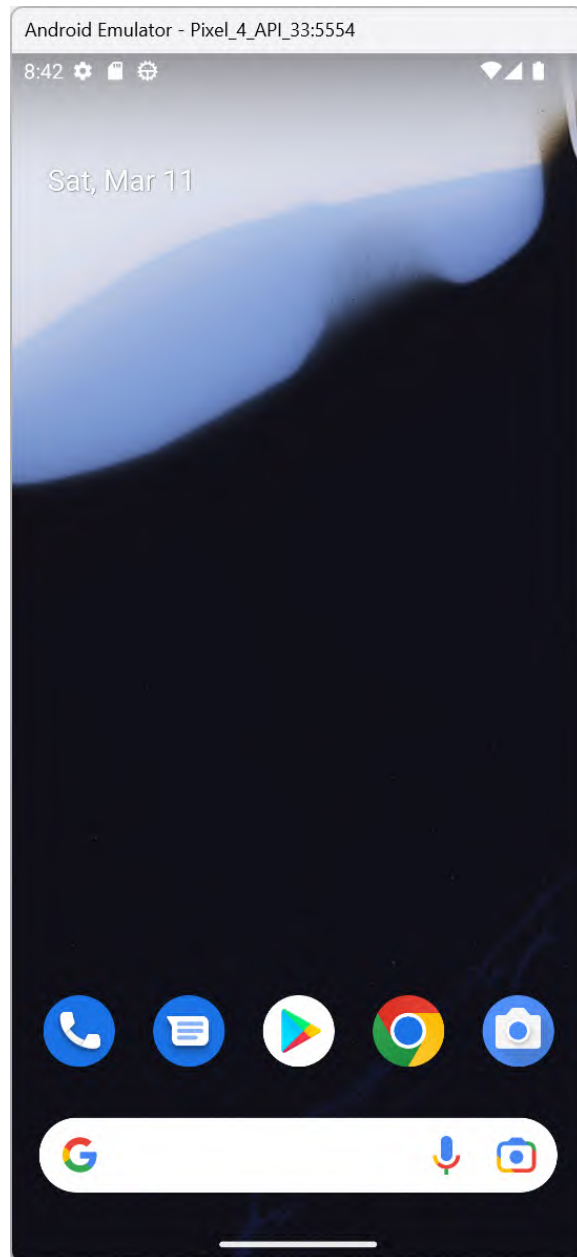


กดปุ่ม Finish

คุณจะได้รายการ Emulator ดังภาพ



กดปุ่ม Play เปิด Emulator ขึ้นมา คุณจะเห็น Emulator ดังภาพ



เปิด terminal สร้าง project ใหม่ด้วยคำสั่ง

```
flutter create helloworld
```

เราจะพบว่า Flutter สร้าง Folder helloworld เพิ่มขึ้นและมีโครงสร้างดังนี้

| Mode | LastWriteTime | Length | Name |
|-------|-------------------|--------|-----------------------|
| d---- | 3/11/2023 8:15 AM | | .dart_tool |
| d---- | 3/11/2023 8:15 AM | | .idea |
| d---- | 3/11/2023 8:15 AM | | android |
| d---- | 3/11/2023 8:15 AM | | ios |
| d---- | 3/11/2023 8:15 AM | | lib |
| d---- | 3/11/2023 8:15 AM | | linux |
| d---- | 3/11/2023 8:15 AM | | macos |
| d---- | 3/11/2023 8:15 AM | | test |
| d---- | 3/11/2023 8:15 AM | | web |
| d---- | 3/11/2023 8:15 AM | | windows |
| -a--- | 3/11/2023 8:15 AM | 745 | .gitignore |
| -a--- | 3/11/2023 8:15 AM | 1668 | .metadata |
| -a--- | 3/11/2023 8:15 AM | 1482 | analysis_options.yaml |
| -a--- | 3/11/2023 8:15 AM | 859 | helloworld.iml |
| -a--- | 3/11/2023 8:15 AM | 5262 | pubspec.lock |
| -a--- | 3/11/2023 8:15 AM | 3964 | pubspec.yaml |
| -a--- | 3/11/2023 8:15 AM | 569 | README.md |

คุณจะพบว่า Flutter สร้างโปรเจกต์ใหม่ พร้อมกับสร้าง Folder สำหรับแอปสำหรับแต่ละแพลตฟอร์มให้ด้วย ได้แก่ android, ios, linux, macos, web และ windows สำหรับการ build ต้องใช้ Toolchain ของแต่ละ platform ในการ build นะ

สำหรับ Folder 'lib' จะเก็บโค้ดของแอป ซึ่งเราจะเขียนโค้ดลงใน Folder นี้ ส่วน pubspec.yaml จะเป็นไฟล์กำหนด dependency และ assets ต่างๆ ที่ใช้ในแอปนี้

การสร้างโปรเจกต์ใหม่ด้วยคำสั่งบน Command Line คุณสามารถระบุ package name และ platform ที่ต้องการได้ เช่น สร้างเฉพาะแอปบน Android และให้ package name เป็น com.mycompany

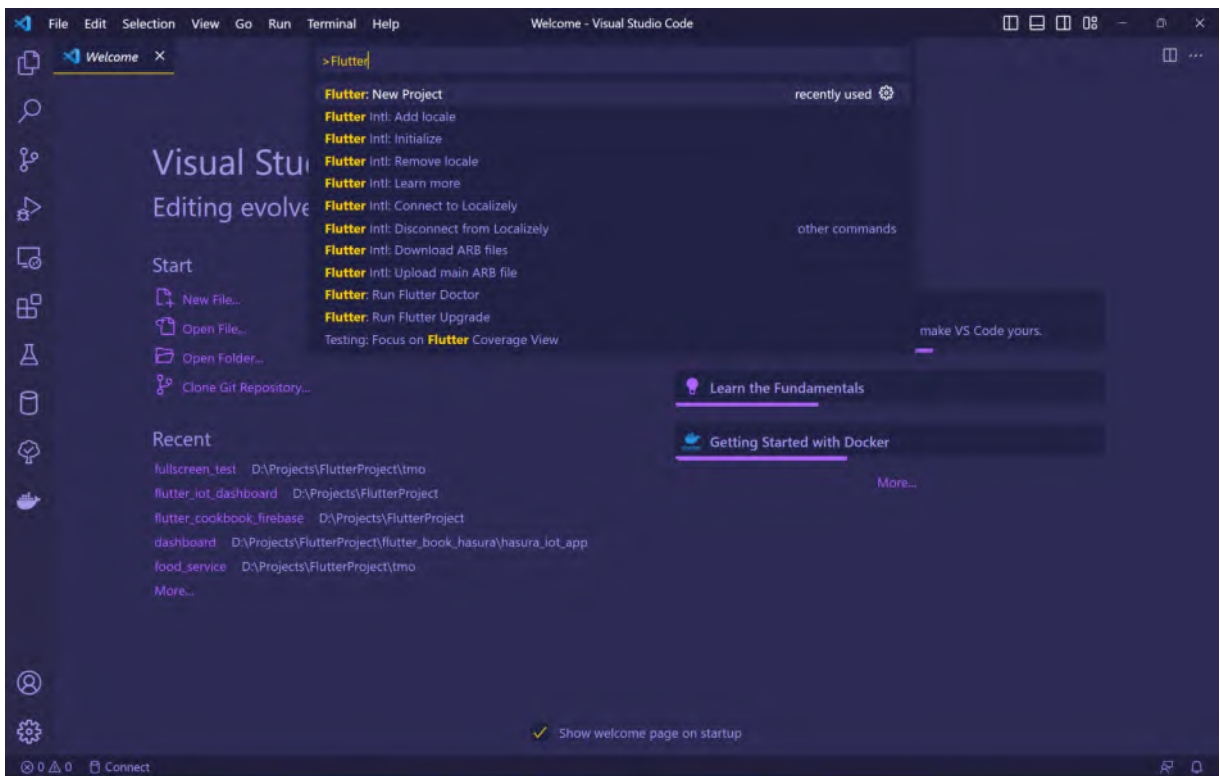
```
flutter create helloworld2 --org com.mycompany
--platforms=android
```

คุณจะพบว่า Folder เฉพาะแพลตฟอร์ม Android ถูกสร้างขึ้นมาเท่านั้น

| Mode | LastWriteTime | Length | Name |
|-------|-------------------|--------|-----------------------|
| d---- | 3/11/2023 8:26 AM | | .dart_tool |
| d---- | 3/11/2023 8:26 AM | | .idea |
| d---- | 3/11/2023 8:26 AM | | android |
| d---- | 3/11/2023 8:26 AM | | lib |
| d---- | 3/11/2023 8:26 AM | | test |
| -a--- | 3/11/2023 8:26 AM | 745 | .gitignore |
| -a--- | 3/11/2023 8:26 AM | 930 | .metadata |
| -a--- | 3/11/2023 8:26 AM | 1482 | analysis_options.yaml |
| -a--- | 3/11/2023 8:26 AM | 859 | helloworld2.iml |
| -a--- | 3/11/2023 8:26 AM | 5262 | pubspec.lock |
| -a--- | 3/11/2023 8:26 AM | 3965 | pubspec.yaml |
| -a--- | 3/11/2023 8:26 AM | 570 | README.md |

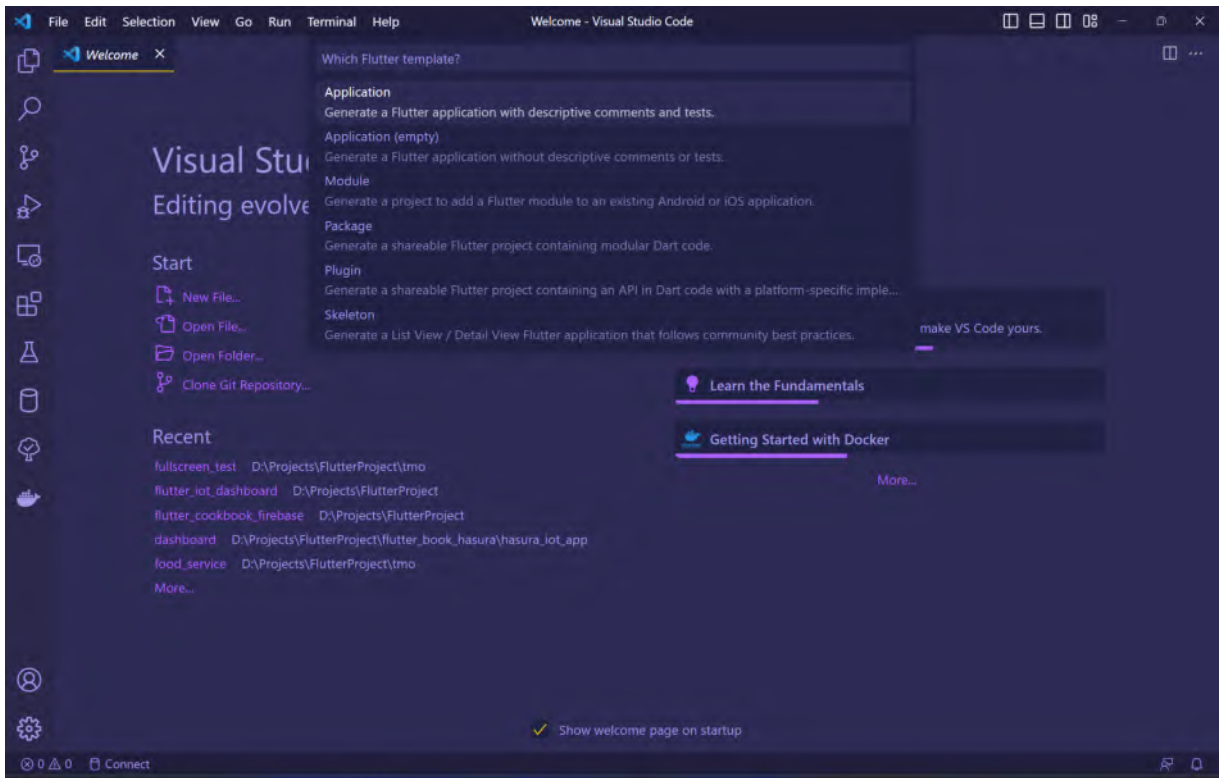
นอกจากนี้คุณยังสามารถใช้ Command Palette ของ VSCode ได้ โดยกด Ctrl+Shift+p บน macOS กด Cmd+Shift+p

แล้วพิมพ์ Flutter : New Project

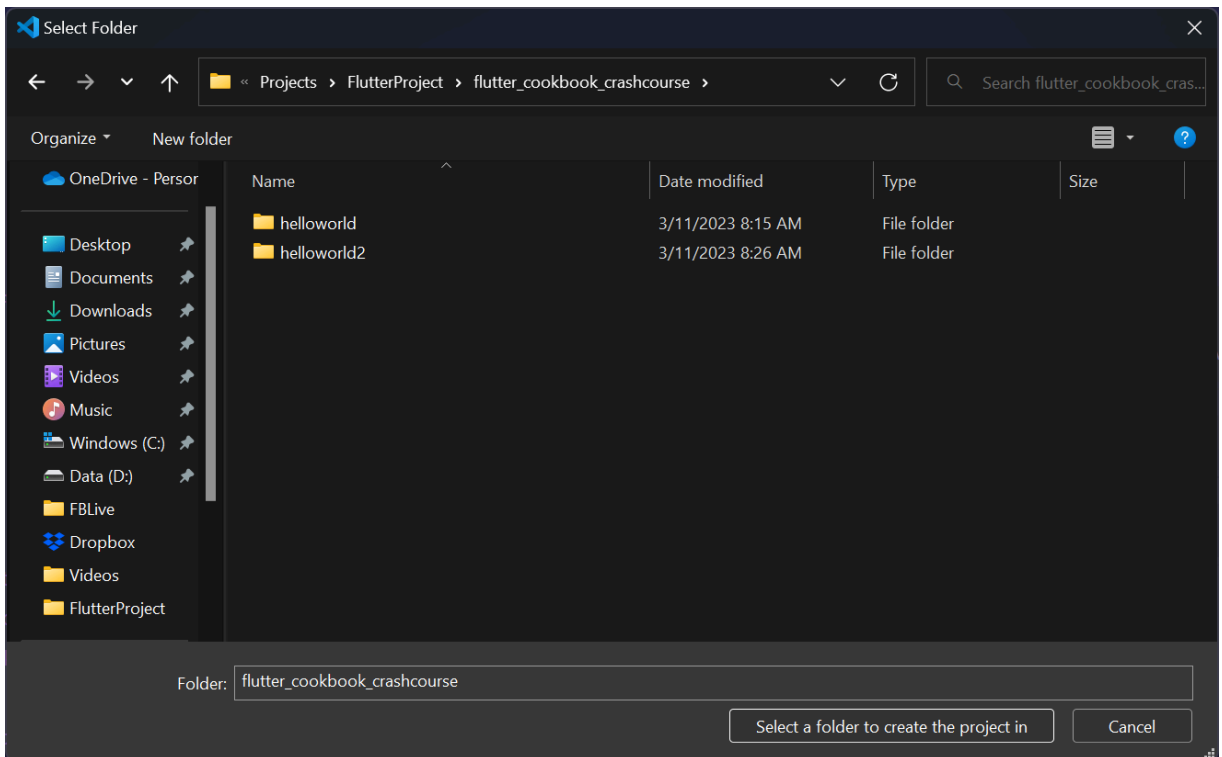


เลือก Flutter : New Project กด Enter

เลือกประเภทของ Project ให้เลือก Application



เลือก Folder สำหรับเก็บไฟล์



ใส่ชื่อ Project ลงไป เช่น helloworld3 Flutter จะสร้างโปรเจคใหม่ให้ คุณจะ
ได้เห็นหน้าจอดังภาพ