

FLUTTER WEEK5

PATIPAN WATJANAPRON

การเชื่อมต่อ API (2)

RESTFUL API

RESTful API Methods



GET

Retrieve a
resource



POST

Create a
resource



PUT

Replace a
resource



PATCH

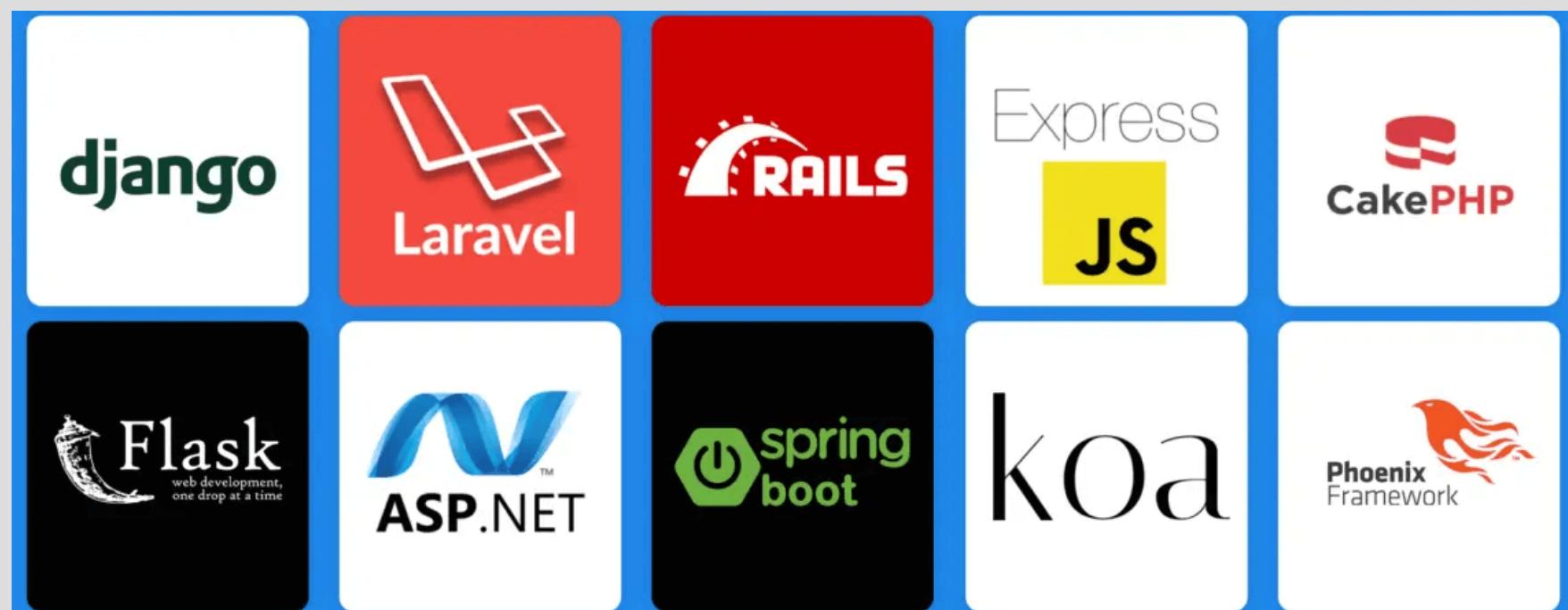
Update a
resource



DELETE

Delete a
resource

BACKEND FRAMEWORK



ตัวอย่าง BACK-END FRAMEWORK



FastAPI

FAST API

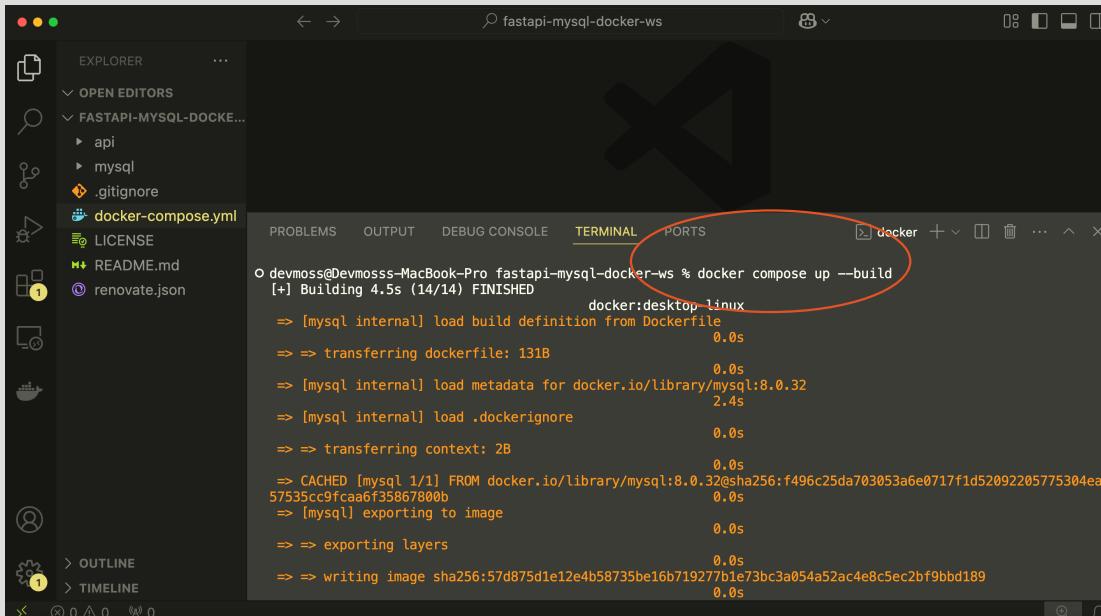
FastAPI พัฒนาด้วย Python เป็นอีกหนึ่งตัวที่น่าสนใจในการนำมาสร้าง API โดยตัวนี้ออกแบบมาให้ง่ายต่อการพัฒนาเป็นอย่างมากเราสามารถที่จะสร้าง API ขึ้นมาได้อย่างรวดเร็ว เข้าใจได้ง่ายและถ้าพูดถึงความเร็วในการใช้งานเชิงประสิทธิภาพนั้นเรียกได้ว่าเร็วเทียบเท่าตัว NodeJS และ Go

INSTALL&BUILD FAST API เพื่อทำ BACK-END

- โหลดไฟล์ fast-api-mysql-docker ที่เตรียมไว้ให้
https://drive.google.com/drive/folders/1aNysgj0jZQa96foLNZhkxcUi85Hi-V2N?usp=drive_link

INSTALL&BUILD FAST API เพื่อทำ BACK-END

- เปิดด้วย VSCode และเปิด Terminal พิมพ์คำสั่งสร้าง container ด้วย
docker compose up -d หรือ docker compose up -build



A screenshot of the Visual Studio Code interface. The left sidebar shows a project structure with folders for 'api', 'mysql', '.gitignore', 'LICENSE', 'README.md', and 'renovate.json'. The right side features a dark-themed terminal window. The terminal tab is active, showing the command 'devmoss@Devmoss-Pro:~/fastapi-mysql-docker-ws % docker compose up --build' followed by a list of build logs. A red oval highlights the terminal tab and the command entered.

```
devmoss@Devmoss-Pro:~/fastapi-mysql-docker-ws % docker compose up --build
[+] Building 4.5s (14/14) FINISHED
          docker:desktop_linux
=> [mysql internal] load build definition from Dockerfile           0.0s
=> > transferring dockerfile: 131B                                0.0s
=> [mysql internal] load metadata for docker.io/library/mysql:8.0.32  2.4s
=> [mysql internal] load .dockerrcignore                         0.0s
=> > transferring context: 28                                    0.0s
=> [mysql internal] FROM docker.io/library/mysql:8.0.32@sha256:f496c25da703053a6e0717f1d52092205775304ea
57535cc9fcfaa6f35867800b                                         0.0s
=> [mysql] exporting to image                                     0.0s
=> > exporting layers                                           0.0s
=> > writing image sha256:57d875d1e12e4b58735be16b719277b1e73bc3a054a52ac4e8c5ec2bf9bbd189   0.0s
```

หน้าจัดการ DB (PHPMYADMIN)

- หน้าจัดการ DB เข้าผ่าน <http://localhost:8040>

The screenshot shows the phpMyAdmin interface at <http://localhost:8040>. The left sidebar shows databases: fastapi_app (selected), information_schema, and performance_schema. The main area shows the 'product' table with the following data:

	<th>name</th> <th>description</th> <th>price</th>	name	description	price
	1	Iphone16 Plus	Apple mobile phone	55900
	2	Samsung S25Ultra	Samsung mobile phone	50900

Below the table, there are buttons for Print, Copy to clipboard, Export, Display chart, and Create view.

- Username : admin
- Password : admin

FAST API (DOCS)

The screenshot shows a web browser window displaying the FastAPI documentation at `localhost:8001/docs/`. The title bar reads "FastAPI - Swagger UI". The main content area is titled "FastAPI 0.1.0 OAS 3.1" and includes a link to `/openapi.json`.

The "default" endpoint is expanded, showing the following operations:

- `GET /` Root
- `GET /products` Read Products
- `POST /products` Create New Product
- `GET /products/{product_id}` Read Product
- `PUT /products/{product_id}` Update Existing Product
- `DELETE /products/{product_id}` Delete Existing Product

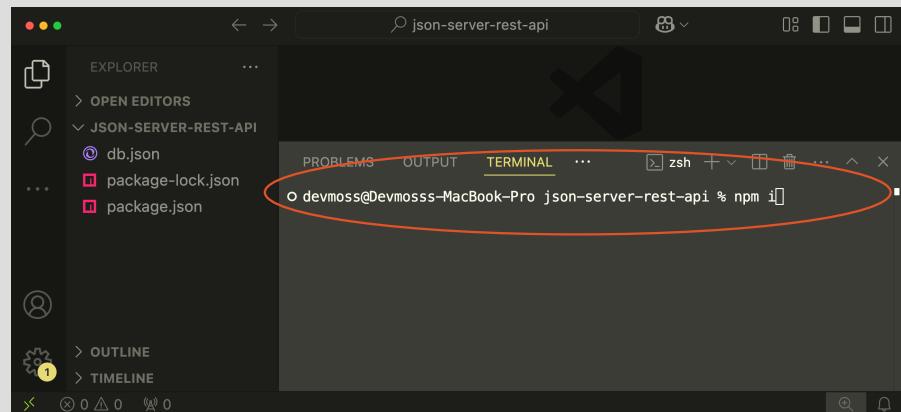
Each operation is color-coded: GET (blue), POST (green), PUT (orange), and DELETE (red). The "Create New Product" operation is highlighted with a green background.

REST API จำลอง (กรณีใช้ DOCKER ไม่ได้)

- สามารถใช้ REST API จำลองจาก File JSON ได้ โดยใช้ lib json-server โดยติดตั้งผ่าน npm install (ต้องใช้ Node)
 - เช็คว่าติดตั้ง Node ในเครื่องหรือยัง node -v
- devmoss@Devmosss-MacBook-Pro json_server % node -v
v18.12.1
- ดาวน์โหลด Node (ถ้ายังไม่มี) <https://nodejs.org/en/download>

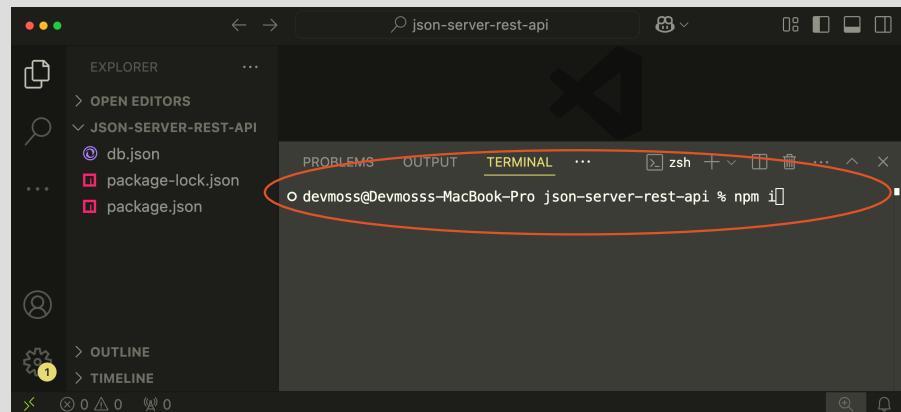
REST API จำลอง (กรณีใช้ DOCKER ไม่ได้)

- โหลดไฟล์ json-server-rest-api ที่เตรียมไว้ให้
https://drive.google.com/drive/folders/1v8mbnj9gmRkplw3aeU_exXDFtlTcsgS?usp=drive_link
- เปิดด้วย VS CODE และเปิด Terminal พิมพ์คำสั่งติดตั้ง lib ด้วย
npm i



REST API จำลอง (กรณีใช้ DOCKER ไม่ได้)

- โหลดไฟล์ json-server-rest-api ที่เตรียมไว้ให้
https://drive.google.com/drive/folders/1v8mbnj9gmRkplw3aeU_exXDFtlTcsgS?usp=drive_link
- เปิดด้วย VS CODE และเปิด Terminal พิมพ์คำสั่งติดตั้ง lib ด้วย
npm i



REST API จำลอง (กรณีใช้ DOCKER ไม่ได้)

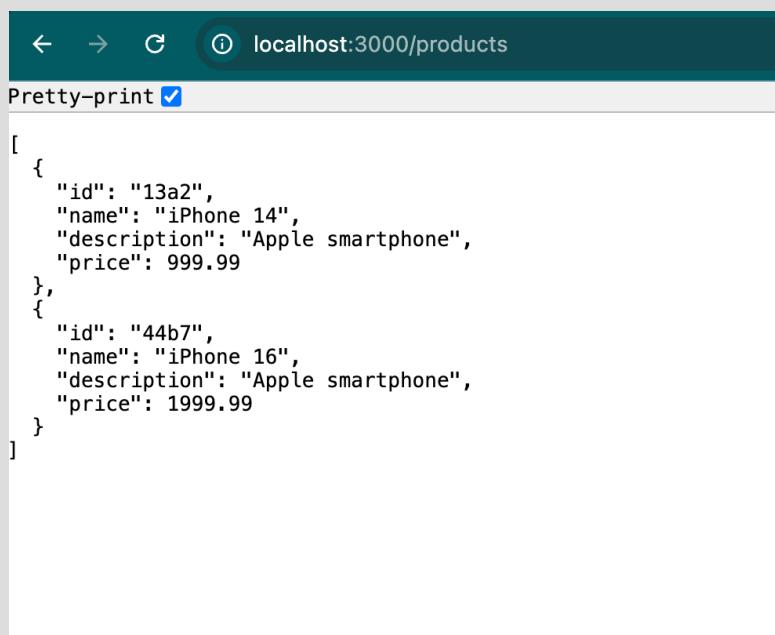
- สั่ง run json-server ด้วยคำสั่ง npx json-server db.json

```
devmoss@Devmoss-MacBook-Pro json-server-rest-api % npx json-server db.json
Need to install the following packages:
  json-server@1.0.0-beta.3
Ok to proceed? (y) y
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: 'milliparsec@4.0.0',
npm WARN EBADENGINE   required: { node: '>=20' },
npm WARN EBADENGINE   current: { node: 'v18.12.1', npm: '8.19.2' }
npm WARN EBADENGINE }
JSON Server started on PORT :3000
Press CTRL-C to stop
Watching db.json...
```

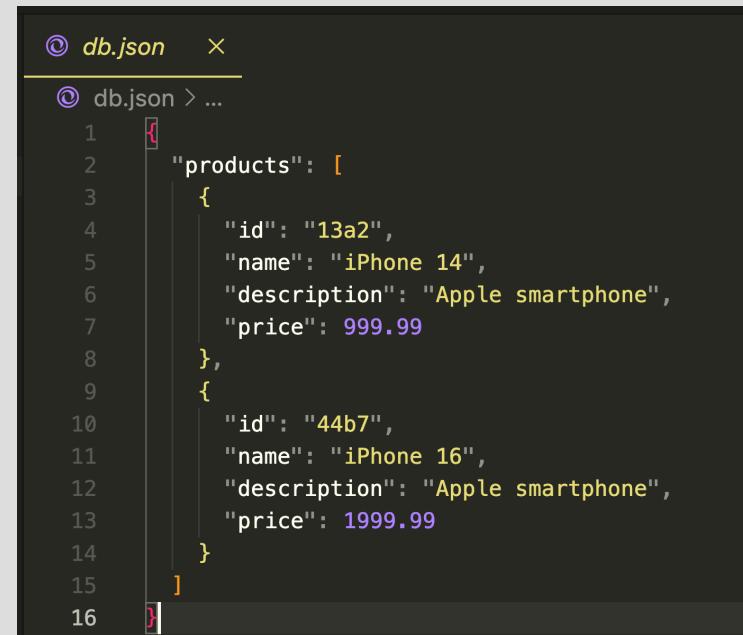
PORT

REST API จำลอง (กรณีใช้ DOCKER ไม่ได้)

- ลองเรียก method get ผ่าน `http://localhost:3000/products`



```
[{"id": "13a2", "name": "iPhone 14", "description": "Apple smartphone", "price": 999.99}, {"id": "44b7", "name": "iPhone 16", "description": "Apple smartphone", "price": 1999.99}]
```



```
1  [
2   "products": [
3     {
4       "id": "13a2",
5       "name": "iPhone 14",
6       "description": "Apple smartphone",
7       "price": 999.99
8     },
9     {
10      "id": "44b7",
11      "name": "iPhone 16",
12      "description": "Apple smartphone",
13      "price": 1999.99
14    }
15  ]
16 ]
```

โปรแกรมสำหรับทดสอบ API (POSTMAN)

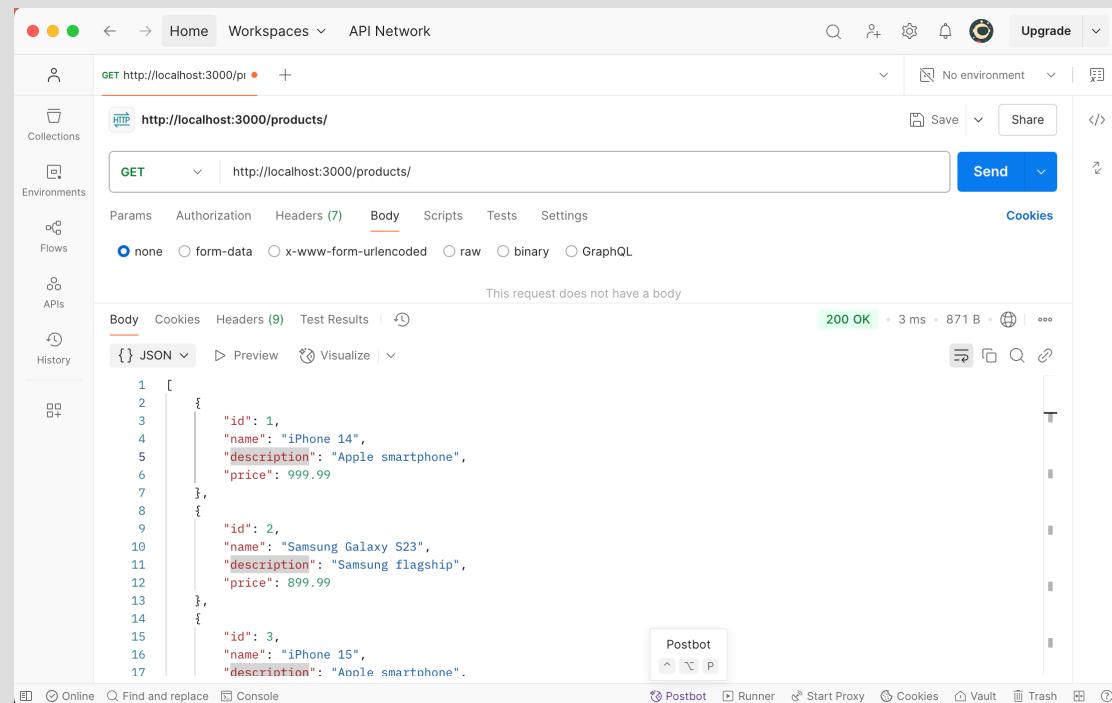
- ดาวน์โหลดโปรแกรม POSTMAN ที่ลิงก์ <https://www.postman.com/downloads/>



POSTMAN

ໂປຣແກຣມສໍາหารັບທດສອບ API (POSTMAN)

- ທດສອບເຮືອກ API ພ່ານ Method GET, PUT, POST, DELETE



การเรียก API ใน LOCAL ผ่าน EMULATOR

- การเรียกใช้งาน local API ผ่าน emulator อาจเกิดปัญหา CORS SITE ได้ โดยในการ Run emulator Android ให้ใช้ <http://10.0.2.2:{port}> และการใช้ <http://localhost:{port}> เช่น <http://10.0.2.2:8001/products>
- ส่วนการ Run emulator ใน Chrome สามารถเรียก <http://localhost:{port}> ได้ตามปกติ เนื่องจาก API ได้ตั้งค่า CORS Site Allow All ไว้แล้ว

FLUTTER HTTP GET (READ)

- การเรียก API ด้วย package http method get

```
Future<void> fetchData() async {
  try {
    var Response response =
        await http.get(Uri.parse(uri: 'http://localhost:3000/products'));
    if (response.statusCode == 200) {
      List<dynamic> jsonList = jsonDecode(source: response.body);
      //code somthing...
    } else {
      throw Exception(message: "Failed to load products");
    }
  } catch (e) {
    print(object: e);
  }
}
```

FLUTTER HTTP POST (CREATE)

- การเรียก API ด้วย package http method post

```
Future<void> createProduct() async {
  try {
    var response = await http.post(
      url: Uri.parse(uri: "http://localhost:3000/products"),
      headers: <String, String>{
        "Content-Type": "application/json",
      },
      body: jsonEncode(object: <String, Object>{
        "name": "iPhone 5s",
        "description": "Apple smartphone",
        "price": 21999.00
      }));
    if (response.statusCode == 201) {
      //code somthing...
    } else {
      throw Exception(message: "Failed to load products");
    }
  } catch (e) {
    print(object: e);
  }
}
```

FLUTTER HTTP PUT (UPDATE)

- การเรียก API ด้วย package http method put

```
Future<void> updateProduct({dynamic idUpdate = "44b7"}) async {
  try {
    var response = await http.put(
      url: Uri.parse(uri: "http://localhost:3000/products/$idUpdate"),
      headers: <String, String>{
        "Content-Type": "application/json",
      },
      body: jsonEncode(object: <String, Object>{
        "name": "iPhone 5 plus",
        "description": "Apple smartphone",
        "price": 34900.00
      }));
    if (response.statusCode == 200) {
      //code somthing...
    } else {
      throw Exception(message: "Failed to load products");
    }
  } catch (e) {
    print(object: e);
  }
}
```

FLUTTER HTTP DELETE

- การเรียก API ด้วย package http method delete

```
Future<void> deleteProduct({dynamic idDelete = "44b7"}) async {
  try {
    var Response response = await http
      .delete(Uri.parse(uri: "http://localhost:3000/products/$idDelete"));
    if (response.statusCode == 200) {
      //code somthing...
    } else {
      throw Exception(message: "Failed to delete products");
    }
  } catch (e) {
    print(object: e);
  }
}
```

SUCCESS AND CONFIRM DIALOG

- การแสดง Snackbar

```
ScaffoldMessenger.of(context: context).showSnackBar(snackBar: const SnackBar(  
    content: Text(data: 'Success!'),  
    backgroundColor: Colors.green,  
) ); // SnackBar
```

SUCCESS AND CONFIRM DIALOG

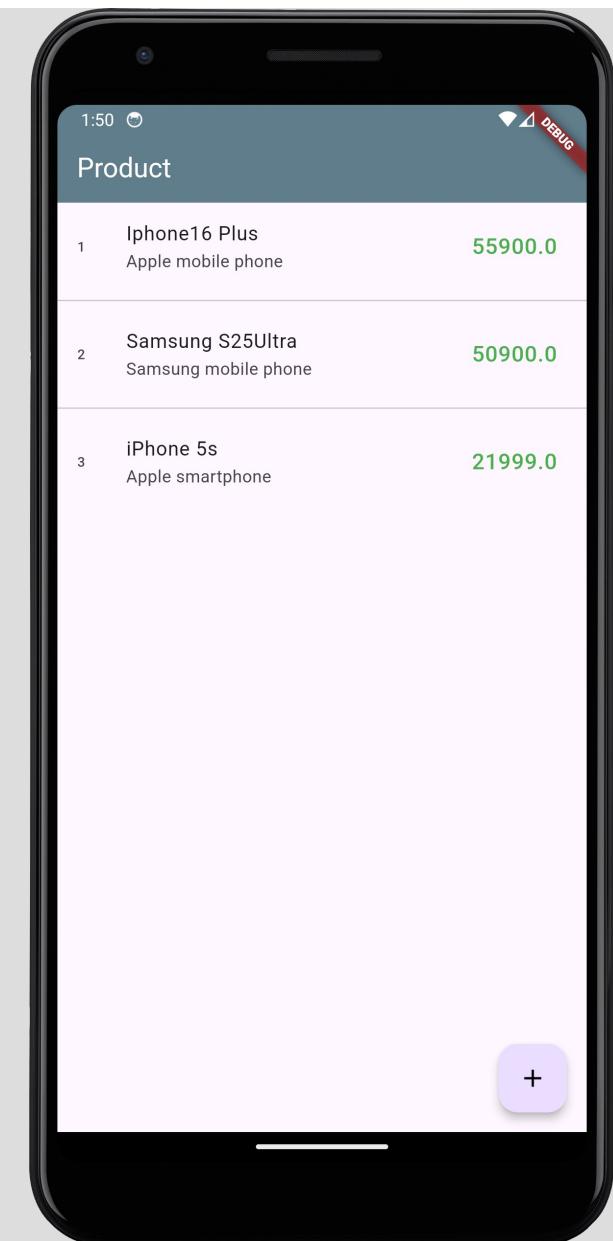
- การแสดง Dialog

```
showDialog<String>(
  context: context,
  builder: (BuildContext context) => AlertDialog(
    title: const Text(data: 'AlertDialog Title'),
    content: const Text(data: 'AlertDialog description'),
    actions: <Widget>[
      TextButton(
        onPressed: () => Navigator.pop<String>(context: context, result: 'Cancel'),
        child: const Text(data: 'Cancel'),
      ), // TextButton
      TextButton(
        onPressed: () => Navigator.pop<String>(context: context, result: 'OK'),
        child: const Text(data: 'OK'),
      ), // TextButton
    ], // <Widget>[]
  ), // AlertDialog
);
```

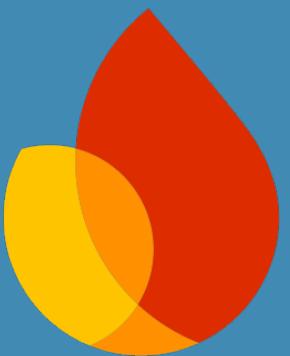
แบบฝึกหัด

โจทย์: ออกรูปแบบแอปพลิเคชันจัดการข้อมูลผ่าน RestAPI

- กำหนดให้ใช้ API จาก FastAPI ที่ build ก่อนหน้านี้ หรือ json-server
- กำหนดให้ออนแอปพลิเคชันมีฟีเจอร์ดังนี้
 - 1.หน้าจอแสดง List รายการสินค้าทั้งหมด
 - 2.หน้าจอ Form สำหรับสร้าง/แก้ไข
 - 3.ปุ่มสำหรับลบข้อมูลสินค้า
 - 4.ต้องแสดง Snackbar หรือ Dialog ตามความจำเป็น เช่นเพิ่มข้อมูลสำเร็จ โชว์ Snackbar ข้อความสำเร็จ หรือเมื่อจะกดลบรายการสินค้า ขึ้น Dialog เพื่อยืนยันว่าต้องการลบข้อมูลนั้นๆ



FIREBASE

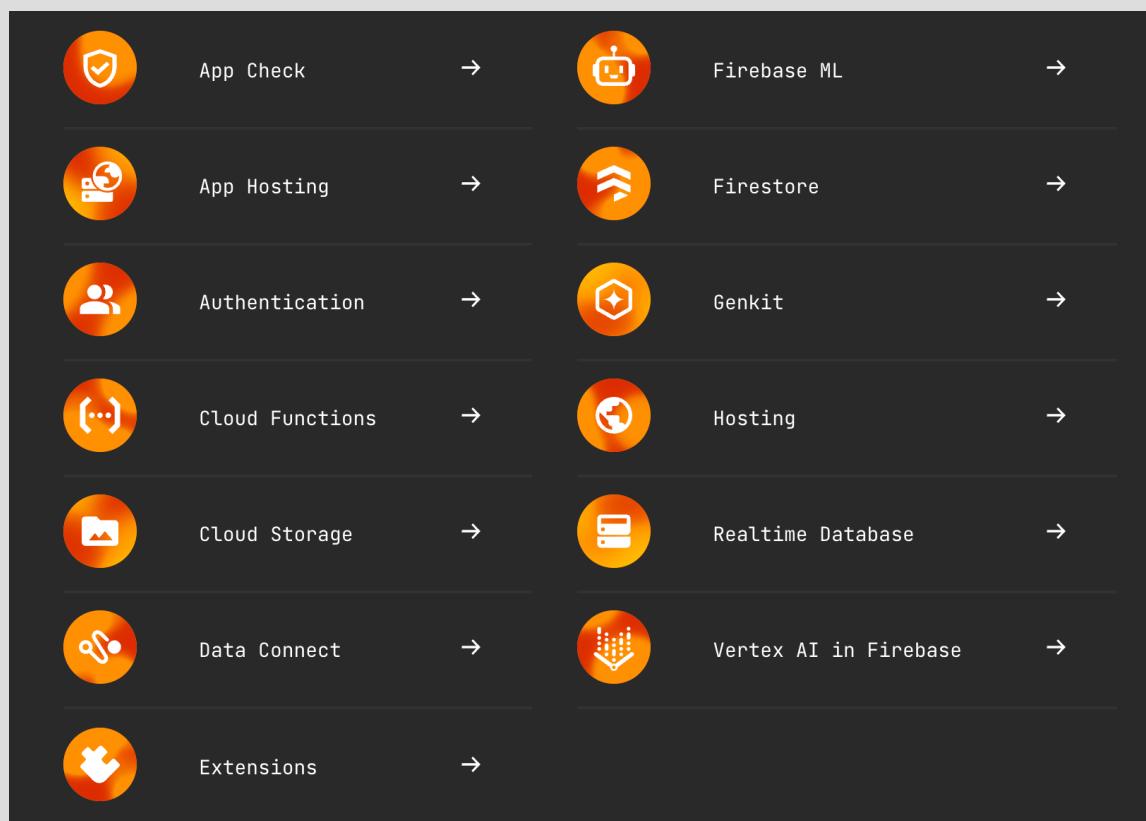


WHAT IS FIREBASE

Firebase คือแพลตฟอร์มพัฒนาแอปพลิเคชันครบวงจร (Mobile and Web Application Development Platform) ซึ่งถูกสร้างขึ้นโดย Firebase Inc. และปัจจุบันอยู่ภายใต้การดูแลของ Google

Firebase ช่วยนักพัฒนาให้ลดเวลาการพัฒนาแอป โดยให้บริการในรูปแบบ **Backend as a Service (BaaS)** หมายความว่า ไม่จำเป็นต้องสร้างเซิร์ฟเวอร์หรือเขียนโค้ดฝั่ง Backend เอง เพราะ Firebase เตรียมบริการพื้นฐานเหล่านี้ไว้ให้แล้ว

CORE SERVICES OF FIREBASE

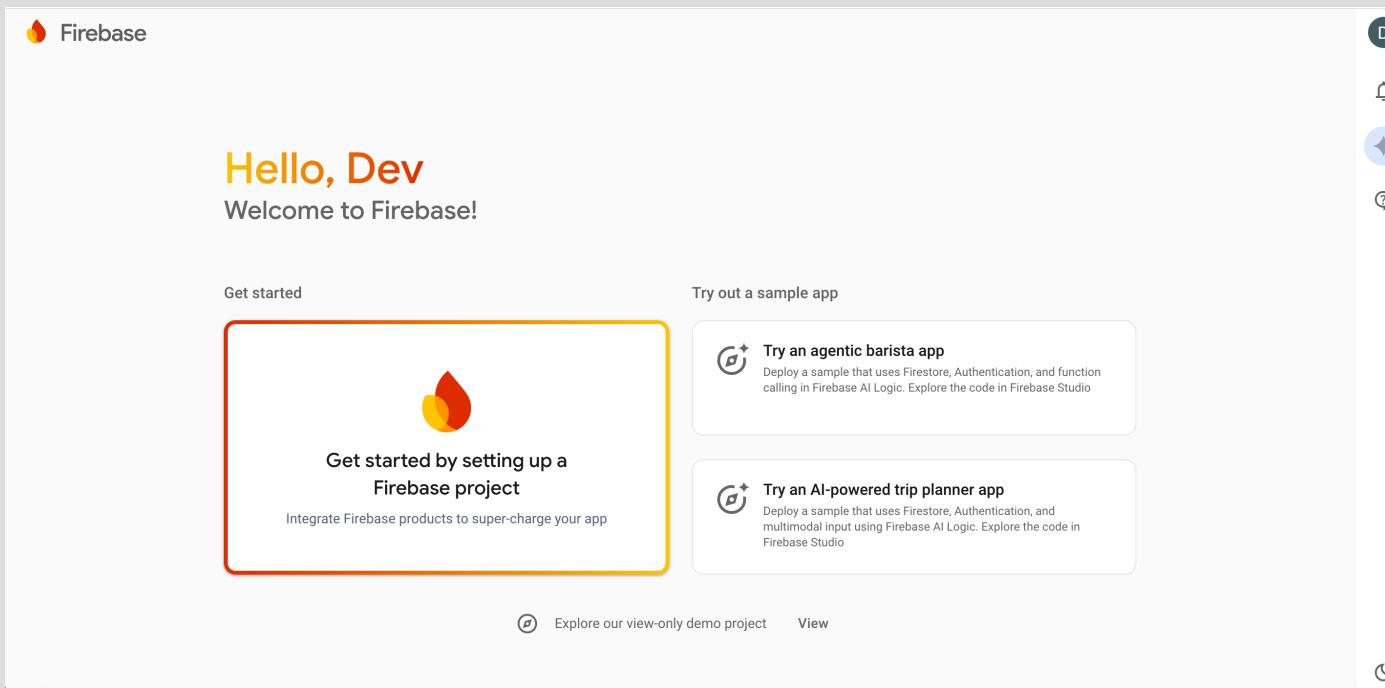


CLOUD FIRESTORE

- Cloud Firestore เป็น ฐานข้อมูล NoSQL บน Cloud ของ Firebase ที่ช่วยให้แอป อ่าน/เขียนข้อมูลแบบเรียลไทม์ และ ซิงค์ข้อมูลระหว่างอุปกรณ์ ได้อย่างมีประสิทธิภาพ
- เหมาะสำหรับ
 - ระบบที่ต้องการข้อมูล เรียลไทม์
 - ระบบที่ต้องการจัดเก็บ JSON-based data

SET UP CLOUD FIRESTORE

- สร้างโปรเจค Firebase ที่ <https://firebase.google.com/>



SET UP CLOUD FIRESTORE

× Create a project

Let's start with a name for
your project^②

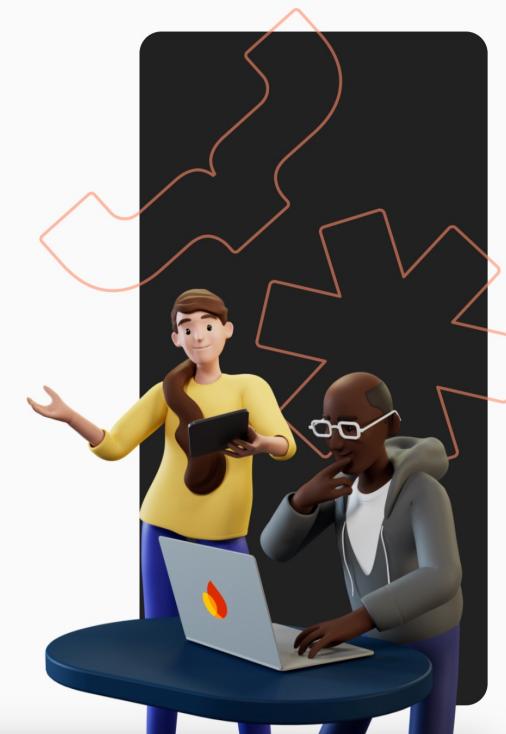
Project name

ProjectStoreDataTest

projectstoredatatest

Already have a Google Cloud project?
[Add Firebase to Google Cloud project](#)

Continue



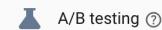
ตั้งชื่อ Project

SET UP CLOUD FIRESTORE

x Create a project

Google Analytics is a free and unlimited analytics solution that enables targeting, reporting, and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, and Cloud Functions.

Google Analytics enables:



A/B testing



Breadcrumb logs in Crashlytics

User segmentation & targeting across Firebase products



Event-based Cloud Functions triggers



Free unlimited reporting

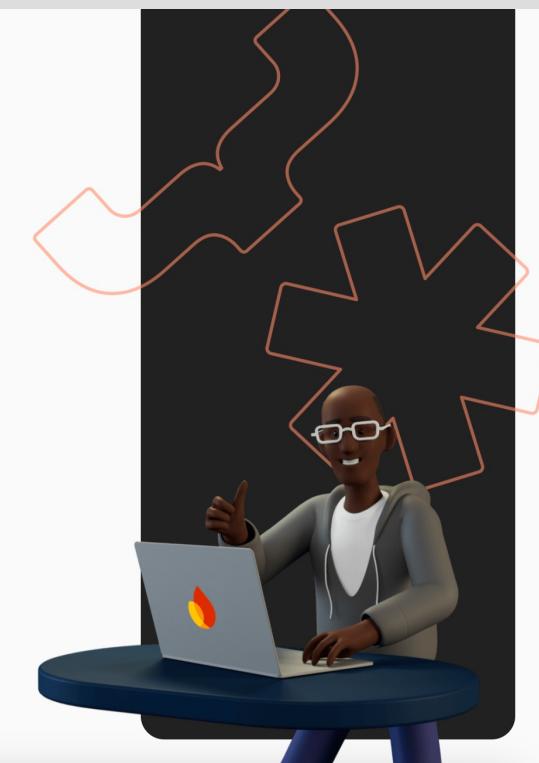


Enable Google Analytics for this project
Recommended

Previous

Continue

ព័ត៌មាន Analytics Project



SET UP CLOUD FIRESTORE

เลือก Analytics location

Configure Google Analytics

Analytics location ⓘ

Thailand

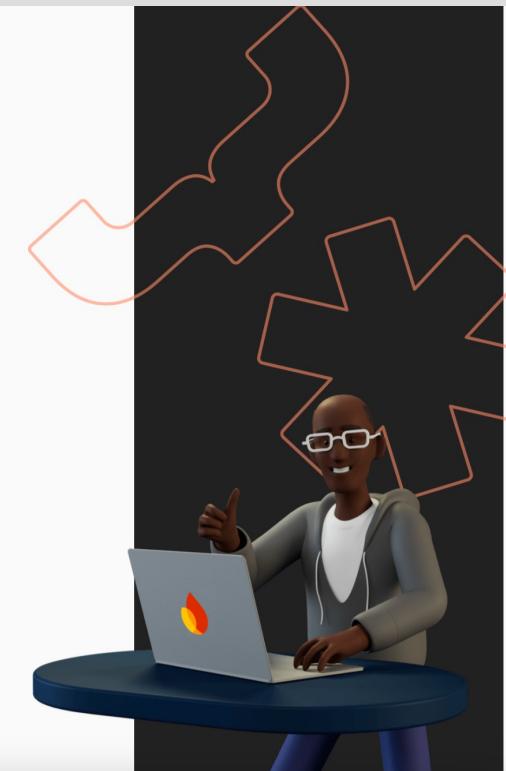
Google Analytics is a business tool. Use it exclusively for purposes related to your trade, business, craft, or profession.

Data sharing settings and Google Analytics terms

Use the default settings for sharing Google Analytics data. [Learn more ↗](#)

- Share your Analytics data with Google to improve Google Products and Services
- Share your Analytics data with Google to enable Benchmarking
- Share your Analytics data with Google to enable Technical Support
- Share your Analytics data with Google Account Specialists

I accept the [Google Analytics terms ↗](#)



SET UP CLOUD FIRESTORE

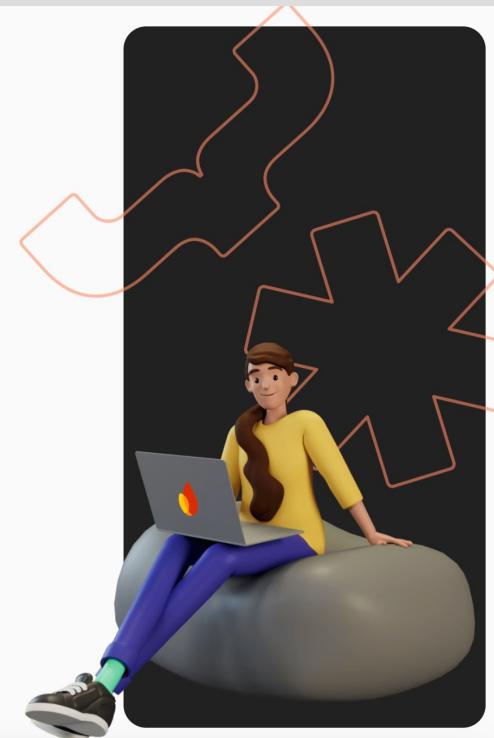
สร้าง Project สำเร็จ



ProjectStoreDataTest

✓ Your Firebase project is ready

Continue



SET UP CLOUD FIRESTORE

ໃໝ່ຈຳ Firestore
database

The screenshot shows the Firebase console interface. On the left, a sidebar menu is open with the following items:

- Build
 - App Check
 - App Hosting
 - Authentication
 - Data Connect
 - Extensions
 - Firebase Database** (this item is highlighted with a red oval)
 - Functions
 - Hosting
 - Machine Learning
 - Realtime Database
 - Storage
- Run
 - Spark (No-cost (\$0/month))
 - Upgrade (NEW)

The main content area is titled "Cloud Firestore" and contains the following text:
Realtime updates, powerful queries, and automatic scaling

Buttons: Create database, Ask Gemini

Illustration: A stack of blue server racks with a magnifying glass focusing on the top one, set against a dark background with a globe icon.

Learn more

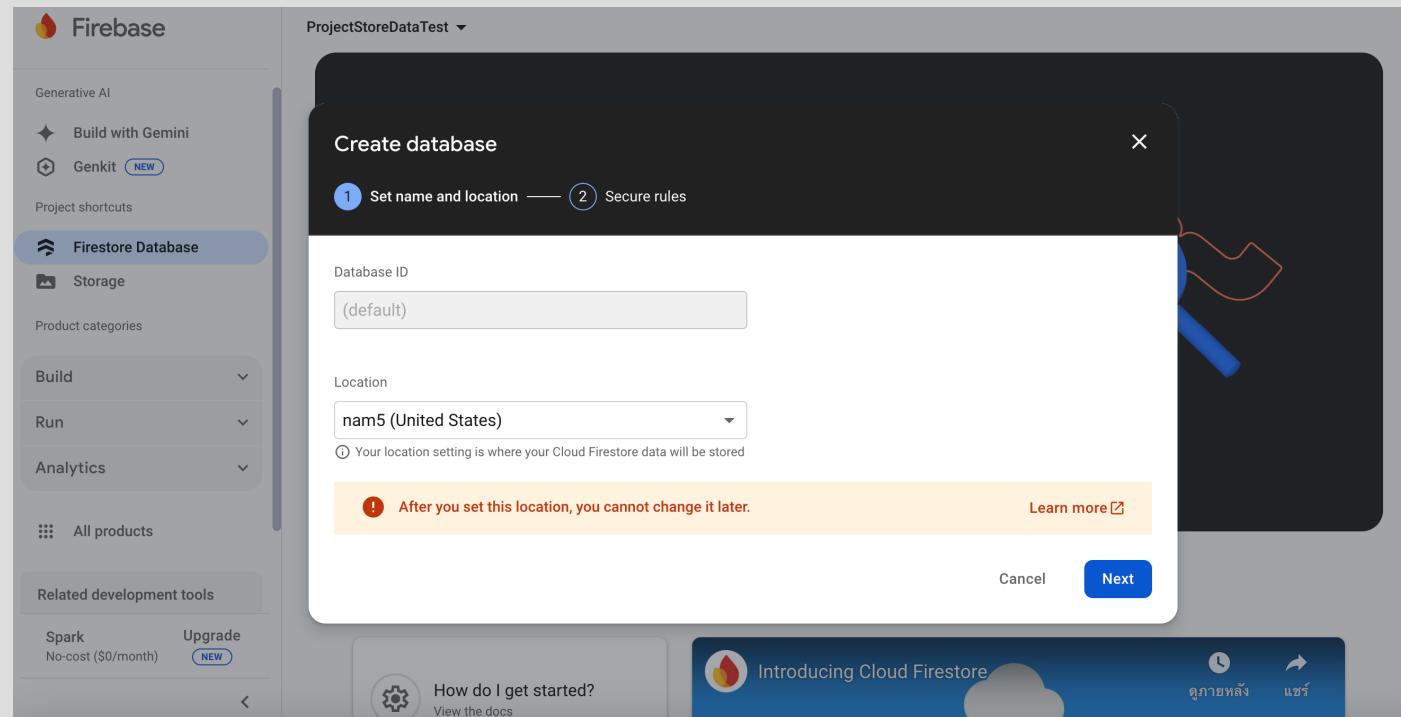
How do I get started? View the docs

Introducing Cloud Firestore

Cloud Firestore Documentation

SET UP CLOUD FIRESTORE

ติดตั้ง lib



SET UP CLOUD FIRESTORE

The screenshot shows the Firebase console interface. On the left, there's a sidebar with options like Generative AI, Build with Gemini, Genkit (NEW), Project shortcuts, Firestore Database (which is selected and highlighted in blue), Storage, Product categories, Build, Run, Analytics, All products, and Related development tools (Spark, Upgrade NEW). A modal window titled "Create database" is open in the center. It has two main steps: "Set name and location" (step 1) and "Secure rules" (step 2). Step 1 is completed. Step 2 is shown with the heading "After you define your data structure, you will need to write rules to secure your data." Below this, there are two radio button options: "Start in production mode" (selected) and "Start in test mode". Under "production mode", it says: "Your data is private by default. Client read/write access will only be granted as specified by your security rules." Under "test mode", it says: "Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long-term client read/write access." To the right of these options is a code editor containing a sample security rule:

```
rules_version = '2';

service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if false;
    }
  }
}
```

Below the code editor, a note states: "All third party reads and writes will be denied". At the bottom of the modal are "Cancel" and "Create" buttons. At the very bottom of the screen, there's a footer bar with icons for "How do I get started?", "View the docs", "Introducing Cloud Firestore", and language selection buttons for "ไทย" and "English".

ຕັ້ງຄ່າ Database

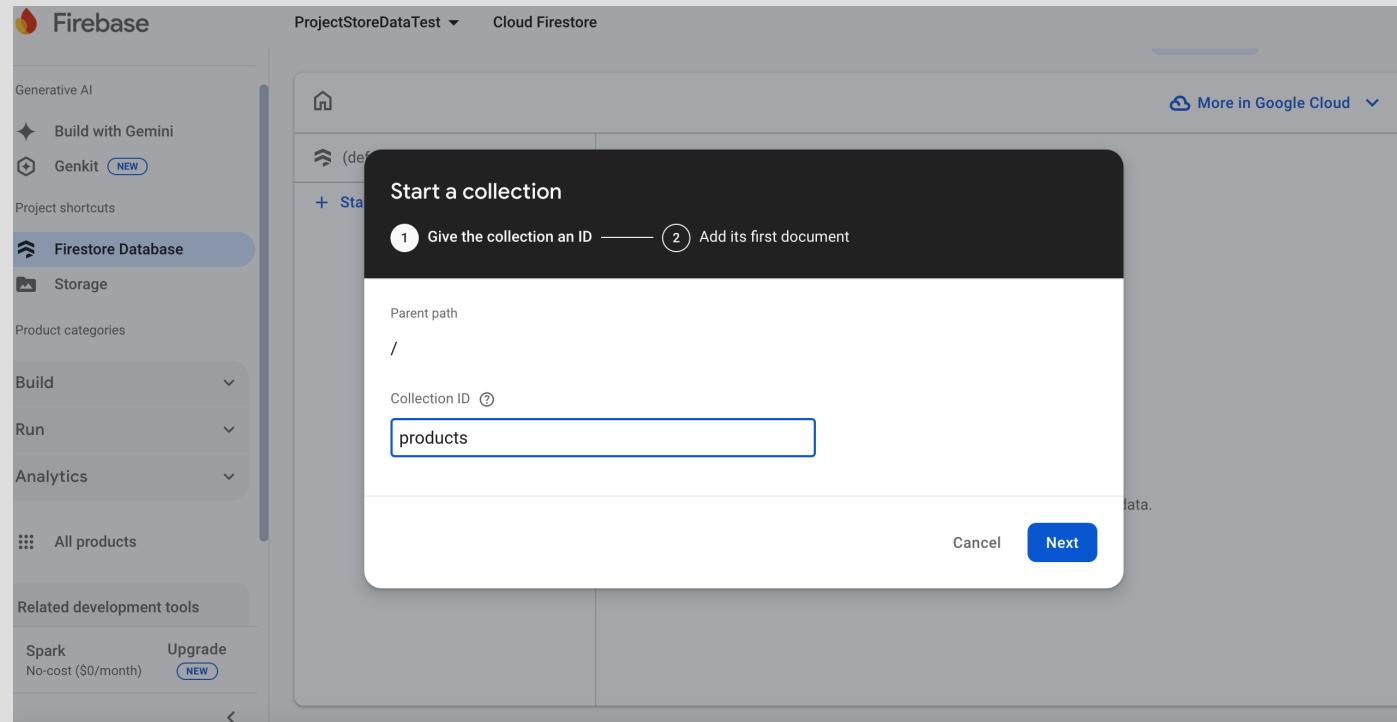
SET UP CLOUD FIRESTORE

The screenshot shows the Firebase Cloud Firestore interface. On the left, a sidebar menu includes 'Generative AI', 'Build with Gemini', 'Genkit (NEW)', 'Project shortcuts', 'Firestore Database' (which is selected and highlighted in blue), 'Storage', 'Product categories', 'Build' (with dropdowns for 'Run' and 'Analytics'), 'All products', and 'Related development tools' (with 'Spark' and 'Upgrade (NEW)'). The main content area displays a collection named '(default)'. Below it is a button labeled '+ Start collection'. A circular icon with a gear and lines is centered in the main area. The text 'Your database is ready to go. Just add data.' is displayed at the bottom. At the very bottom of the page, there is a footer note: 'Database location: nam5'.

พร้อมสำหรับใช้งาน

SET UP CLOUD FIRESTORE

สร้าง collection



SET UP CLOUD FIRESTORE

สร้าง Document
ตัวอย่าง

The screenshot shows the Firebase console interface for setting up a Cloud Firestore collection. A modal window titled 'Start a collection' is open, prompting the user to 'Give the collection an ID' (which is set to '/products') and 'Add its first document'. The document ID is 'TY82auvwqvhBD6d3rubX'. The modal displays three fields: 'name' (string type, value 'Iphone15pro'), 'description' (string type, value 'Apple mobile'), and 'price' (string type, value '30900'). A blue box highlights the 'price' field.

SET UP CLOUD FIRESTORE

แสดง Collections และ
Documents ที่มี

The screenshot shows the Firebase Cloud Firestore interface. On the left, the navigation bar includes 'Generative AI', 'Build with Gemini', 'Genkit (NEW)', 'Project shortcuts', and 'Firestore Database' (which is selected and highlighted in blue). Below these are 'Storage', 'Product categories', 'Build', 'Run', 'Analytics', and 'All products'. Under 'Related development tools', there are 'Spark' (No-cost (\$0/month)) and 'Upgrade (NEW)'. The main workspace shows a hierarchical view: Home > products > TY82auvwqvhB... . The 'products' collection contains one document, 'TY82auvwqvhBD6d3rubX', which is expanded to show its fields: 'description: "Apple mobile"', 'name: "Iphone15pro"', and 'price: "30900"'. A 'More in Google Cloud' dropdown menu is visible at the top right.

SET UP CLOUD FIRESTORE

Allow read & write

The screenshot shows the Google Cloud Platform interface for Cloud Firestore. On the left, there's a sidebar with icons for Home, Settings, and a plus sign. The main area has tabs for Data, Rules (which is selected), Indexes, Disaster Recovery (NEW), Usage, and Extensions. A button labeled "Ask Gemini how to get started with Firestore" is visible. On the right, there's a "Develop & Test" button and a vertical toolbar with icons for Home, Data, Rules, Indexes, and Help.

The central part of the screen displays a "Rules Playground" section. It shows a list of documents on the left: "Today • 5:40 PM", "Today • 5:30 PM", "Today • 5:24 PM", and "Today • 2:40 PM". To the right is a code editor with the following security rules:

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /products/{docId} {
      allow read, write;
    }
  }
}
```

A large red arrow points from the text "Allow read & write" in the bottom-left corner towards the "allow read, write;" line in the code editor.

SET UP CLOUD FIRESTORE

ติดตั้ง SDK กับโปรเจค

The screenshot shows the Firebase Project Overview page for a project named "ProjectStoreDataTest". The left sidebar includes links for Project Overview, Generative AI (Build with Gemini, Genkit), Project shortcuts (Firestore Database, Storage), Product categories, Build, Run, Analytics, All products, Spark (No-cost (\$0/month)), and Upgrade (NEW). The main area displays the "Environment" settings with "Unspecified" selected. Below that is the "Your apps" section, which says "There are no apps in your project" and prompts to "Select a platform to get started". It features icons for iOS+, Android (circled in red with a green arrow pointing to it), Web, and Cloud Functions. A "Delete project" button is at the bottom right.

เลือกติดตั้งกับ Android

SET UP CLOUD FIRESTORE

copy package id
.../android/app/
build.gradle



```
EXPLORER ... build.gradle X
OPEN EDITORS
DEMPROJECT
  .dart_tool
  .idea
  android
    .gradle
    app
      src
        debug
        main
        profile
        build.gradle
        .gradle
        build.gradle
        demoproject_android...
        gradle.properties
        gradlew
        gradlew.bat
        local.properties
        settings.gradle
        assets
        ...
        > OUTLINE

  android > app > build.gradle
  android {
    namespace = "com.example.demoproject"
    compileSdk = flutter.compileSdkVersion
    ndkVersion = flutter.ndkVersion
    compileOptions {
      sourceCompatibility = JavaVersion.VERSION_1_8
      targetCompatibility = JavaVersion.VERSION_1_8
    }
    kotlinOptions {
      jvmTarget = JavaVersion.VERSION_1_8
    }
  }
  defaultConfig {
    // TODO: Specify your own unique Application ID (https://developer.android.com/studio)
    applicationId = "com.example.demoproject"
    // You can update the following values to match your application needs.
    // For more information, see: https://flutter.dev/to-review-gradle-config.
    minSdk = flutter.minSdkVersion
    targetSdk = flutter.targetSdkVersion
    versionCode = flutter.versionCode
  }
```

SET UP CLOUD FIRESTORE

[Go to docs](#)

× Add Firebase to your Android app

1 Register app

Android package name [?](#)

com.example.demoproject

App nickname (optional) [?](#)

Demoproject

Debug signing certificate SHA-1 (optional) [?](#)

① Required for Dynamic Links, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.

Register app

2 Download and then add config file

SET UP CLOUD FIRESTORE

ติดตั้ง service.json
ตามคำแนะนำ

- 1 Register app
Android package name: com.example.demoproject, App nickname: Demoproject
- 2 Download and then add config file

[Download google-services.json](#)

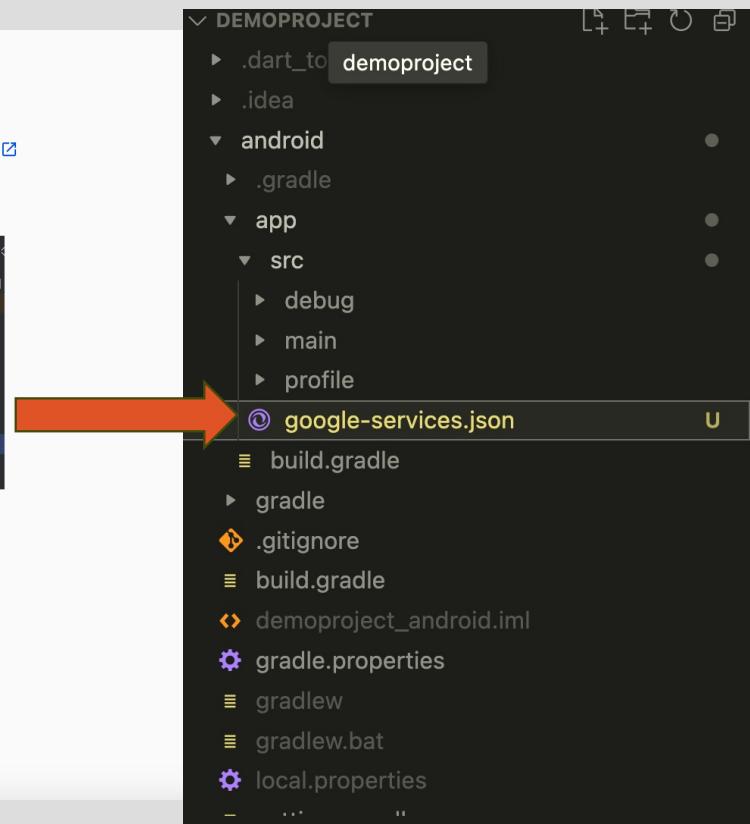
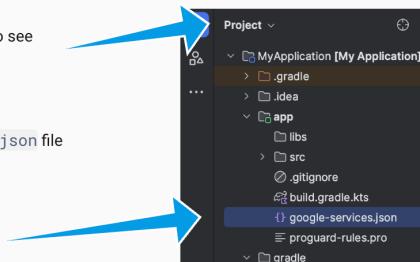
Switch to the Project view in Android Studio to see your project root directory.

Move your downloaded `google-services.json` file into your module (app-level) root directory.



Next

- 3 Add Firebase SDK
- 4 Next steps



SET UP CLOUD FIRESTORE

ติดตั้ง service.json
ตามคำแนะนำ

1 Register app
Android package name: com.example.demoproject, App nickname: Demoproject

2 Download and then add config file
Instructions for Android Studio below | [Unity](#) [C++](#)

[Download google-services.json](#)

Switch to the Project view in Android Studio to see your project root directory.

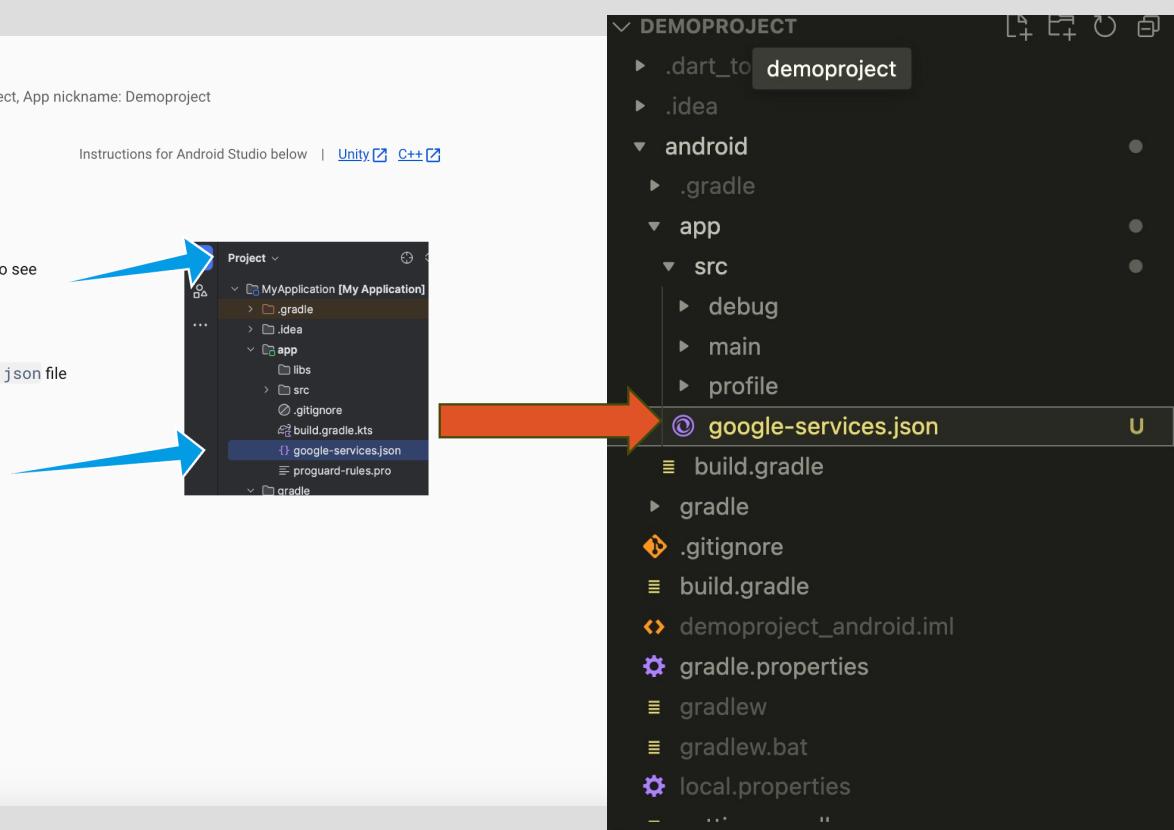
Move your downloaded `google-services.json` file into your module (app-level) root directory.

`google-services.json`

Next

3 Add Firebase SDK

4 Next steps



SET UP CLOUD FIRESTORE

ขั้น step นี้



★ Are you still using the buildscript syntax to manage plugins? Learn how to add Firebase plugins using that syntax.

1. To make the `google-services.json` config values accessible to Firebase SDKs, you need the Google services Gradle plugin.

Kotlin DSL (`build.gradle.kts`) Groovy (`build.gradle`)

Add the plugin as a dependency to your `project-level build.gradle` file:

`Root-level (project-level) Gradle file (<project>/build.gradle):`

```
plugins {
    ...
    // Add the dependency for the Google services Gradle plugin
    id 'com.google.gms.google-services' version '4.4.2' apply false
}
```

2. Then, in your `module (app-level) build.gradle` file, add both the `google-services` plugin and any Firebase SDKs that you want to use in your app:

`Module (app-level) Gradle file (<project>/<app-module>/build.gradle):`

```
plugins {
    id 'com.android.application'
    // Add the Google services Gradle plugin
    id 'com.google.gms.google-services'
    ...
}

dependencies {
    // Import the Firebase BoM
    implementation platform('com.google.firebase:firebase-bom:33.9.0')

    // TODO: Add the dependencies for Firebase products you want to use
    // When using the BoM, don't specify versions in Firebase dependencies
    implementation 'com.google.firebaseio:firebase-analytics'

    // Add the dependencies for any other desired Firebase products
    // https://firebase.google.com/docs/android/setup#available-libraries
}
```

By using the Firebase Android BoM, your app will always use compatible Firebase library versions. [Learn more](#)

3. After adding the plugin and the desired SDKs, sync your Android project with Gradle files.

SET UP CLOUD FIRESTORE

เพิ่ม Firebase SDK
ไฟล์ android/app/build.gradle

```
android > app > build.gradle
  1 1   plugins {
  2 2     id "com.android.application"
  3 3     id "kotlin-android"
  4 4     // The Flutter Gradle Plugin must be applied after the Android and Kotlin Gradle plugins.
  5 5     id "dev.flutter.flutter-gradle-plugin"
  6 6+    id "com.google.gms.google-services" // នេះ plugin រួច
  7 7 }
  8 8
  9 9   android {
 10 10     namespace = "com.example демопроект"
 11 11     compileSdk = flutter.compileSdkVersion
 12 12     ndkVersion = flutter.ndkVersion
 13 13
 14 14     compileOptions {
 15 15       sourceCompatibility = JavaVersion.VERSION_1_8
 16 16       targetCompatibility = JavaVersion.VERSION_1_8
 17 17   }
 18 18
 19 19     kotlinOptions {
 20 20       jvmTarget = JavaVersion.VERSION_1_8
 21 21   }
 22 22
 23 23     defaultConfig {
 24 24       // TODO: Specify your own unique Application ID (https://developer.android.com/studio/build/applicationId)
 25 25       applicationId = "com.example демопроект"
 26 26       // You can update the following values to match your application needs.
 27 27       // For more information, see: https://flutter.dev/to/review-gradle-config.
 28 28       minSdk = flutter.minSdkVersion
 29 29       targetSdk = flutter.targetSdkVersion
 30 30       versionCode = flutter.versionCode
 31 31       versionName = flutter.versionName
 32 32   }
 33 33
 34 34     buildTypes {
 35 35       release {
 36 36         // TODO: Add your own signing config for the release build.
 37 37         // Signing with the debug keys for now, so 'flutter run --release' works.
 38 38         signingConfig = signingConfigs.debug
 39 39     }
 40 40 }
```

SET UP CLOUD FIRESTORE

ติดตั้ง Firebase CLI
พิมใน cmd or terminal
(ต้องมี node ในเครื่องแล้ว)

```
npm i -g firebase-tools
```

SET UP CLOUD FIRESTORE

เข้าสู่ระบบ firebase พิมพ์ใน
cmd or terminal

firebase login # ล็อกอิน Google

SET UP CLOUD FIRESTORE

ดู project ที่มี

firebase projects:list

✓ Preparing the list of your Firebase projects

Project Display Name	Project ID	Project Number	Resource Location ID
first-project	first-project-c242e	974438171183	[Not specified]
test-project	test-project-2cdb1	364098231925	[Not specified]

SET UP CLOUD FIRESTORE

ติดตั้ง flutterfire_cli

```
dart pub global activate flutterfire_cli
```

SET UP CLOUD FIRESTORE

ตรวจสอบ flutterfire_cli
ว่าถูกติดตั้ง และเห็น Path แล้ว
หรือไม่



flutterfire -v

ถ้า flutterfire_cli ใช้งานได้
ปกติ จะแสดงเวอร์ชันได้ปกติ
ให้ข้ามขั้นตอนการ
set path flutterfire



```
devmoss@Devmosss-MacBook-Pro firebase_mini_project % flutterfire -v  
1.3.1
```

ប័ណ្ណហា PATH FLUTTERFIRE នៃ WINDOWS

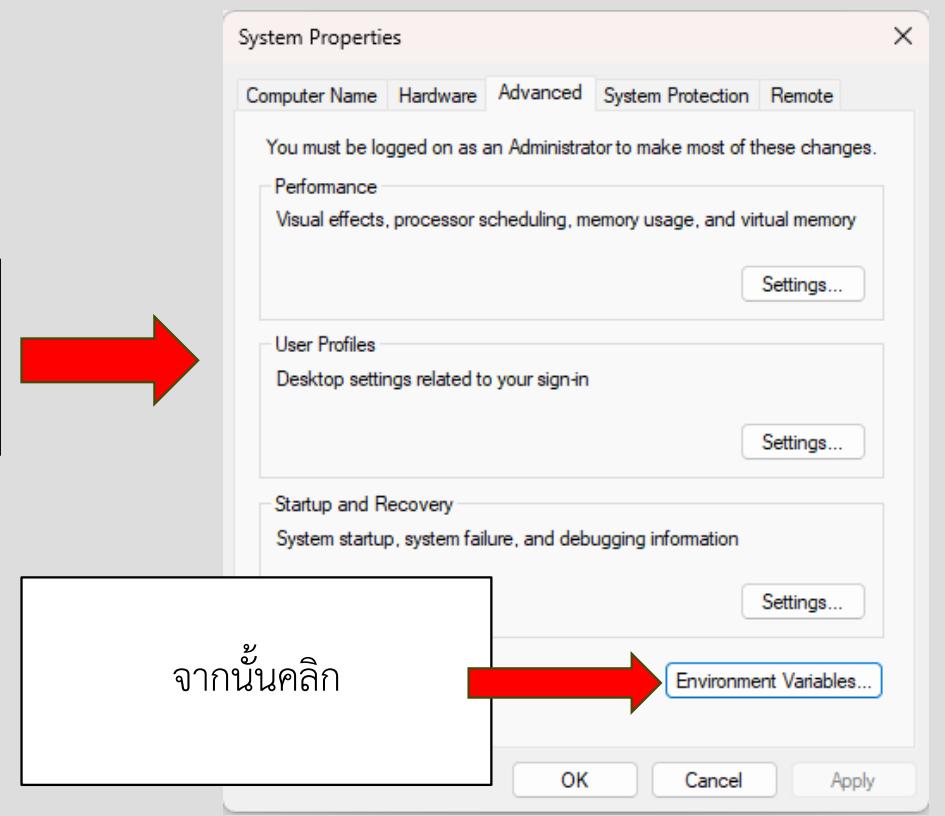
ឬប័ណ្ណហា មិនមែន path

```
C:\Users\ODIN SON>flutterfire -v  
'flutterfire' is not recognized as an internal or external command,  
operable program or batch file.
```

ប័ណ្ណហា PATH FLUTTERFIRE នៃ WINDOWS

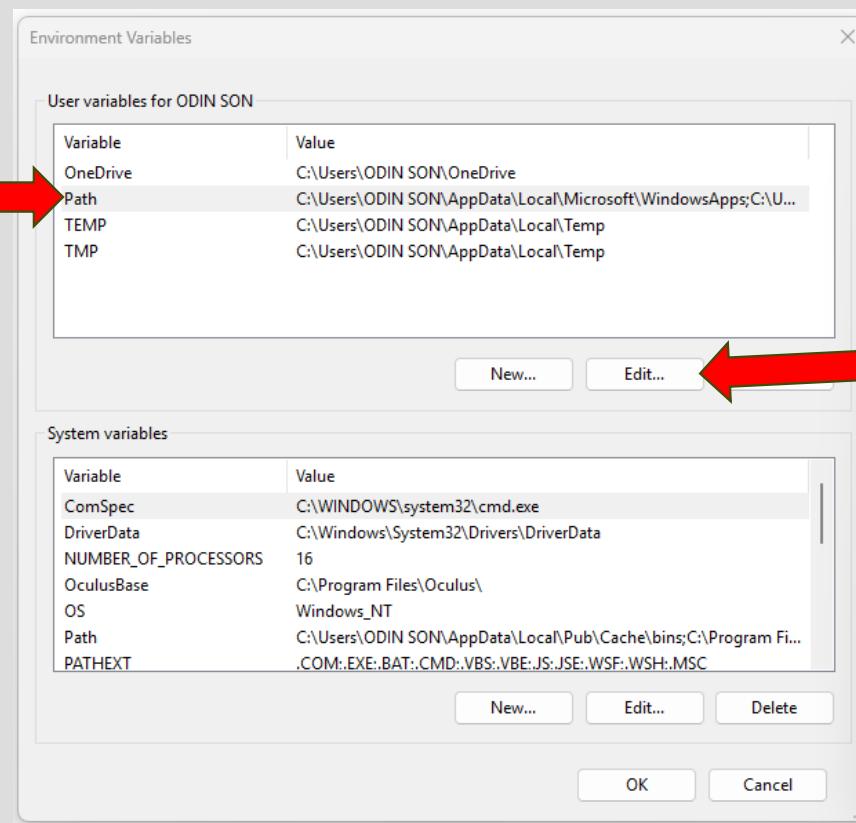
កែតាមការដើរ path លើ
System environment

បើក System Properites
នៃ windows ដើម្បី set path



ប័ណ្ណហា PATH FLUTTERFIRE នៃ WINDOWS

តើ
ឡើក path



ចាកនេះឡើក Edit

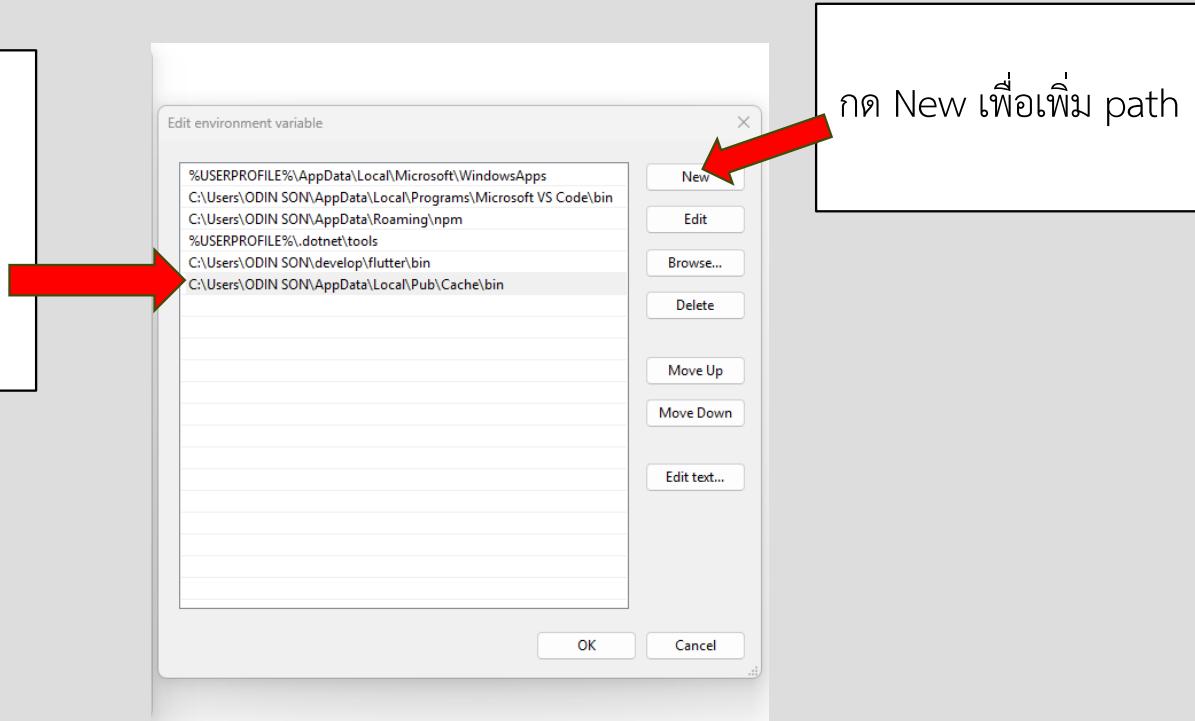
ปัญหา PATH FLUTTERFIRE ใน WINDOWS

เพิ่ม PATH ดังนี้

C:\Users*username*\AppData\Local\Pub\Cache\bin

หมายเหตุ

- *username* คือชื่อ user เครื่อง
- เมื่อ set path เสร็จให้ปิดเปิด cmd หรือ Terminal ใหม่ และลองพิม flutterfire -v อีกครั้ง



ปัญหา PATH FLUTTERFIRE ใน MAC OS

เพิ่ม PATH ใน terminal ดังนี้



```
export PATH="$PATH":$HOME/.pub-cache/bin"
```

SET UP CLOUD FIRESTORE

รัน config Firebase

```
flutterfire configure --platforms=android,ios,web
```

เลือก โปรเจค Firebase ที่จะใช้
ต้องตรงกับไฟล์ package.json

```
i Found 2 Firebase projects.  
? Select a Firebase project to configure your Flutter application with >  
  > first-project-c242e (first-project)  
  test-project-2cdb1 (test-project)  
  <create a new project>
```

SET UP CLOUD FIRESTORE

ติดตั้ง lib cloud_firestore

flutter pub add cloud_firestore

ติดตั้ง lib firebase_core

flutter pub add firebase_core

CLOUD FIRESTORE

```
import 'firebase_options.dart';

Run | Debug | Profile
void main() async {
    WidgetsFlutterBinding.ensureInitialized();
    await Firebase.initializeApp(
        options: DefaultFirebaseOptions.currentPlatform,
    );

    runApp(const MyApp());
}
```

เรียกใช้ Firebase ในไฟล์
main.dart

CLOUD FIRESTORE

Read data (stream)
Cloud Firestore

```
final CollectionReference products =
    FirebaseFirestore.instance.collection('products');

@Override
Widget build(BuildContext context) {
    return Scaffold(
        body: StreamBuilder<QuerySnapshot>(
            stream: products.snapshots(),
            builder: (context, snapshot) {
                if (!snapshot.hasData) {
                    return const CircularProgressIndicator();
                }
                final data = snapshot.data!.docs;
                return ListView.separated(
                    separatorBuilder: (BuildContext context, int index) =>
                        const Divider(),
                    itemCount: data.length,
                    itemBuilder: (context, index) {
                        final product = data[index];
                        String productId = product.id; // ←
                        return ListTile(
                            leading: Text(productId),
                            title: Text(product['name']),
                            subtitle: Text(product['description']),
                        ); // ListTile
                    },
                ); // ListView.separated
            },
        ), // StreamBuilder
    ); // Scaffold
```

Stream Widget
สำหรับอ่านข้อมูลแบบ realtime

ดึง Id ของ document

CLOUD FIRESTORE

Add data to
Cloud Firestore

```
//CREATE: Add New Data
Future<void> createProduct(
    String name, String description, double price) async {
  await products.add({
    'name': name,
    'description': description,
    'price': price,
  });
}
```

CLOUD FIRESTORE

Update data
Cloud Firestore

```
//Update
Future<void> updateProduct(
    String docId, String name, String description, double price) async {
  await products.doc(docId).update({
    'name': name,
    'description': description,
    'price': price,
  });
}
```

CLOUD FIRESTORE

Delete data
Cloud Firestore

```
//Delete
Future<void> deleteProduct(String docId) async {
  await products.doc(docId).delete();
}
```