

Flutter Basic Google Maps 2025



การเขียนแอปพลิเคชันเพื่อแสดงแผนที่ด้วย Google Map สามารถกำหนดการแสดงผลแผนที่ไปยังตำแหน่งของสถานที่ต่าง ๆ ได้โดยการเรียกใช้ package ไลบรารี `google_maps_flutter` ที่ช่วยให้สามารถเรียกใช้งาน Google Map ที่สามารถใช้ในการค้นหาสถานที่ด้วยการกำหนดพิกัดและแผนที่ได้อีกด้วย ซึ่งข้อสำคัญในการใช้งานแผนที่นี้มีข้อสำคัญเกี่ยวกับการเข้าถึงบริการข้อมูลและสิทธิ์ในการใช้ข้อมูลพิกัดจากอุปกรณ์เคลื่อนที่ ดังนั้นแอปพลิเคชันจะต้องถูกตรวจสอบและควบคุม

ดังนั้นการใช้งานจะต้องลงทะเบียนเพื่อสร้าง API KEY สำหรับการใช้งานของผู้พัฒนาและแอปพลิเคชันนั้น ๆ

ขั้นตอนการสร้าง API KEY ไปที่ <https://console.cloud.google.com/> click ที่ NEW PROJECT

The screenshot shows the Google Cloud Platform interface. A modal dialog box titled "Select a resource" is open in the foreground. In the top right corner of this dialog, there is a red rectangular box highlighting the "NEW PROJECT" button. The background shows the main Google Cloud dashboard with sections for "Dashboard", "Recommendations", and "Cloud Storage". On the left, there is a sidebar with "Quick access" and "API APIs & Services". The central area displays a list of recent projects, with "nitiganmap" and "kmutnb.ac.th" listed under the "RECENT" tab.

ตั้งชื่อ project (** มีการจำกัด จำนวนของการสร้าง project)

New Project

Project name * **Flutter Google Map 2024** ?

Project ID: white-gate-416416. It cannot be changed later. [EDIT](#)

Organization * **kmtnb.ac.th** ?

Select an organization to attach it to a project. This selection can't be changed later.

Location * **kmtnb.ac.th** BROWSE

Parent organization or folder

CREATE **CANCEL**

สร้าง api key โดย click ที่ Credentials และ เลือกสร้างที่ CREATE CREDENTIALS

API Credentials - APIs & Services

Flutter Google Map 2024

API APIs & Services + CREATE CREDENTIALS DELETE RESTORE DELETED CREDENTIALS

Enabled APIs & services

Library

Credentials

Create credentials to access your enabled APIs. [Learn more](#)

⚠ Remember to configure the OAuth consent screen with information about your application. [CONFIGURE CONSENT SCREEN](#)

API Keys

Name	Creation date	Restrictions	Actions
No API keys to display			

OAuth 2.0 Client IDs

Name	Creation date	Type	Client ID	Actions
No OAuth clients to display				

Service Accounts

Email	Name	Actions
No service accounts to display		

Screenshot of the Google Cloud API Credentials page showing the creation of an API key.

The page title is "API Credentials - APIs & Services". The URL is <https://console.cloud.google.com/apis/credentials?project=white-gate-416416>.

The left sidebar shows "Enabled APIs & services", "Library", "Credentials" (selected), "OAuth consent screen", and "Page usage agreements".

The main area shows "Credentials" with options: "+ CREATE CREDENTIALS" (highlighted with a red box), "DELETE", and "RESTORE DELETED CREDENTIALS".

A dropdown menu is open under "+ CREATE CREDENTIALS" with the following options:

- API key**: Identifies your project using a simple API key to check quota and access.
- OAuth client ID**: Requests user consent so your app can access the user's data.
- Service account**: Enables server-to-server, app-level authentication using robot accounts.
- Help me choose**: Asks a few questions to help you decide which type of credential to use.

Buttons for "CONFIGURE CONSENT SCREEN" and "RESTRICTIONS" are visible.

Screenshot of the Google Cloud API Credentials page showing the creation of an API key in progress.

The page title is "API Credentials - APIs & Services". The URL is <https://console.cloud.google.com/apis/credentials?project=white-gate-416416>.

The left sidebar shows "Enabled APIs & services", "Library", "Credentials" (selected), "OAuth consent screen", and "Page usage agreements".

The main area shows "Credentials" with options: "+ CREATE CREDENTIALS" (highlighted with a red box), "DELETE", and "RESTORE DELETED CREDENTIALS".

A modal window titled "Creating API key..." is displayed, showing a loading icon.

The background shows the same API Keys section as the first screenshot, with the "Help me choose" option selected.

The screenshot shows the Google Cloud Platform API & Services Credentials page. A modal dialog box is centered, displaying the message "API key created". It contains the text "Use this key in your application by passing it with the `key=API_KEY` parameter." Below this, a section titled "Your API key" shows the value "AIzaSyCs2u9p7R9THYP9qcxE-3wzaVfCT_kfapM". A red box highlights this key value. At the bottom of the modal is a "CLOSE" button.

ให้ Copy Key เก็บไว้สำหรับใช้ใน Code ของ Flutter

The screenshot shows the Google Cloud Platform API & Services Credentials page. The "API Keys" section is highlighted with a red box. It lists one key: "API key 1". The "Actions" column for this key also has a red box around it. Other sections like "OAuth 2.0 Client IDs" and "Service Accounts" are visible below.

จากนั้น Click ที่ API Key เพื่อกำหนดค่าให้กับ API

ກຳທັນດີ API restrictions

API APIs & Services

Enabled APIs & services

Library

Credentials

OAuth consent screen

Page usage agreements

Set an application restriction

Application restrictions limit an API key's usage to specific websites, IP addresses, Android applications, or iOS applications. You can set one application restriction per key.

None

Websites

IP addresses

Android apps

iOS apps

API restrictions

API restrictions specify the enabled APIs that this key can call

Don't restrict key
This key can call any API

Restrict key

Select APIs

Selected APIs:

Note: It may take up to 5 minutes for settings to take effect

SAVE CANCEL

Check Maps SDK Android ແລະ click SAVE

API APIs & Services

Enabled APIs & services

Library

Credentials

OAuth consent screen

Page usage agreements

Set an application restriction

Application restrictions limit an API key's usage to specific websites, IP addresses, Android applications, or iOS applications. You can set one application restriction per key.

None

Websites

IP addresses

Android apps

iOS apps

API restrictions

API restrictions specify the enabled APIs that this key can call

Don't restrict key
This key can call any API

Restrict key

Filter Type to filter

S

Maps SDK for Android

CANCEL OK

API Credentials - APIs & Services - + https://console.cloud.google.com/apis/credentials?project=white-gate-416416

Google Cloud Flutter Google Map 2024 Search (/) for resources, docs, products, and more Search

API APIs & Services Credentials + CREATE CREDENTIALS DELETE RESTORE DELETED CREDENTIALS

Enabled APIs & services Create credentials to access your enabled APIs. [Learn more](#)

Library Remember to configure the OAuth consent screen with information about your application. [CONFIGURE CONSENT SCREEN](#)

Credentials

OAuth consent screen

Page usage agreements

API Keys

<input type="checkbox"/>	Name	Creation date	Restrictions	Actions
<input checked="" type="checkbox"/>	API key 1	Mar 6, 2024	Maps SDK for Android ...	SHOW KEY ⋮

OAuth 2.0 Client IDs

<input type="checkbox"/>	Name	Creation date	Type	Client ID	Actions
No OAuth clients to display					

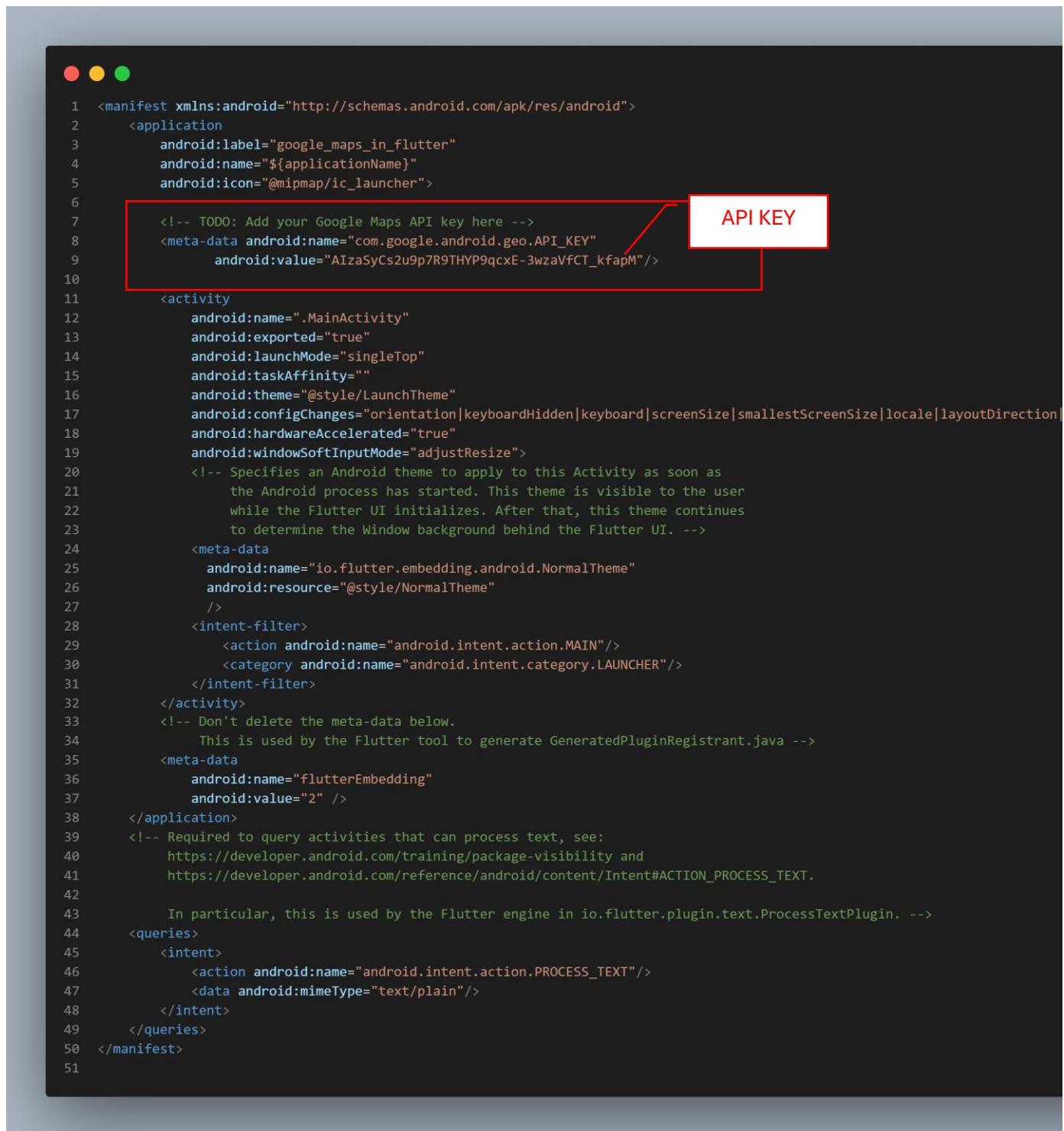
Service Accounts

<input type="checkbox"/>	Email	Name	Actions
No service accounts to display			

[Manage service accounts](#)

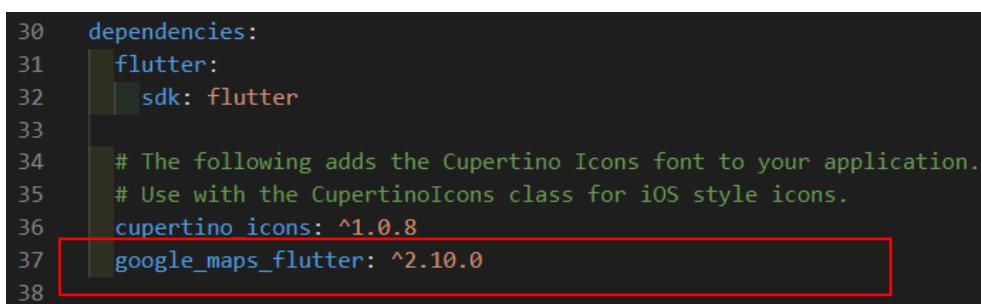
การเขียนแอปพลิเคชัน สำหรับ Android ให้เข้าไปเพิ่ม API Key ที่ไฟล์ AndroidManifest.xml

\android\app\src\main\AndroidManifest.xml



```
1 <manifest xmlns:android="http://schemas.android.com/apk/res/android">
2   <application
3     android:label="google_maps_in_flutter"
4     android:name="${applicationName}"
5     android:icon="@mipmap/ic_launcher">
6
7     <!-- TODO: Add your Google Maps API key here -->
8     <meta-data android:name="com.google.android.geo.API_KEY"
9       android:value="AIzaSyCs2u9p7R9THYP9qcxE-3wzaVfCT_kfapM"/>
10
11    <activity
12      android:name=".MainActivity"
13      android:exported="true"
14      android:launchMode="singleTop"
15      android:taskAffinity=""
16      android:theme="@style/LaunchTheme"
17      android:configChanges="orientation|keyboardHidden|keyboard|screenSize|smallestScreenSize|locale|layoutDirection|fontScale|density|uiMode"
18      android:hardwareAccelerated="true"
19      android:windowSoftInputMode="adjustResize">
20      <!-- Specifies an Android theme to apply to this Activity as soon as
21          the Android process has started. This theme is visible to the user
22          while the Flutter UI initializes. After that, this theme continues
23          to determine the Window background behind the Flutter UI. -->
24      <meta-data
25        android:name="io.flutter.embedding.android.NormalTheme"
26        android:resource="@style/NormalTheme"
27      />
28      <intent-filter>
29        <action android:name="android.intent.action.MAIN"/>
30        <category android:name="android.intent.category.LAUNCHER"/>
31      </intent-filter>
32    </activity>
33    <!-- Don't delete the meta-data below.
34        This is used by the Flutter tool to generate GeneratedPluginRegistrant.java -->
35    <meta-data
36      android:name="flutterEmbedding"
37      android:value="2" />
38  </application>
39  <!-- Required to query activities that can process text, see:
40      https://developer.android.com/training/package-visibility and
41      https://developer.android.com/reference/android/content/Intent#ACTION_PROCESS_TEXT.
42
43      In particular, this is used by the Flutter engine in io.flutter.plugin.text.ProcessTextPlugin. -->
44  <queries>
45    <intent>
46      <action android:name="android.intent.action.PROCESS_TEXT"/>
47      <data android:mimeType="text/plain"/>
48    </intent>
49  </queries>
50 </manifest>
```

ติดตั้ง google_maps_flutter ที่ pubspec.yaml



```
30  dependencies:
31    flutter:
32      sdk: flutter
33
34    # The following adds the Cupertino Icons font to your application.
35    # Use with the CupertinoIcons class for iOS style icons.
36    cupertino_icons: ^1.0.8
37    google_maps_flutter: ^2.10.0
38
```

กำหนด SDK Version ที่ไฟล์ android/app/build.gradle โดยกำหนด minSdkVersion เป็น 21

```
1  plugins {
2      id("com.android.application")
3      id("kotlin-android")
4      // The Flutter Gradle Plugin must be applied after the Android and Kotlin Gradle plugins.
5      id("dev.flutter.flutter-gradle-plugin")
6  }
7
8  android {
9      namespace = "com.example.google_maps_in_flutter"
10     compileSdk = flutter.compileSdkVersion
11     ndkVersion = flutter.ndkVersion
12
13    compileOptions {
14        sourceCompatibility = JavaVersion.VERSION_11
15        targetCompatibility = JavaVersion.VERSION_11
16    }
17
18    kotlinOptions {
19        jvmTarget = JavaVersion.VERSION_11.toString()
20    }
21
22    defaultConfig {
23        // TODO: Specify your own unique Application ID
24        // (https://developer.android.com/studio/build/application-id.html).
25        applicationId = "com.example.google_maps_in_flutter"
26        // You can update the following values to match your application needs.
27        // For more information, see: https://flutter.dev/to/review-gradle-config.
28        minSdk = 21
29        targetSdk = flutter.targetSdkVersion
30        versionCode = flutter.versionCode
31        versionName = flutter.versionName
32    }
33
34    buildTypes {
35        release {
36            // TODO: Add your own signing config for the release build.
37            // Signing with the debug keys for now, so `flutter run --release` works.
38            signingConfig = signingConfigs.getByName("debug")
39        }
40    }
41 }
42
43 flutter {
44     source = "../../"
45 }
46
```

เริ่มต้นการสร้าง Application

import Library

```
import 'package:google_maps_flutter/google_maps_flutter.dart';
```

สร้างตัวแปรประเภท GoogleMapController คือ class ที่เอาไว้สร้าง Future อีกที เอาไว้ควบคุมการทำงาน

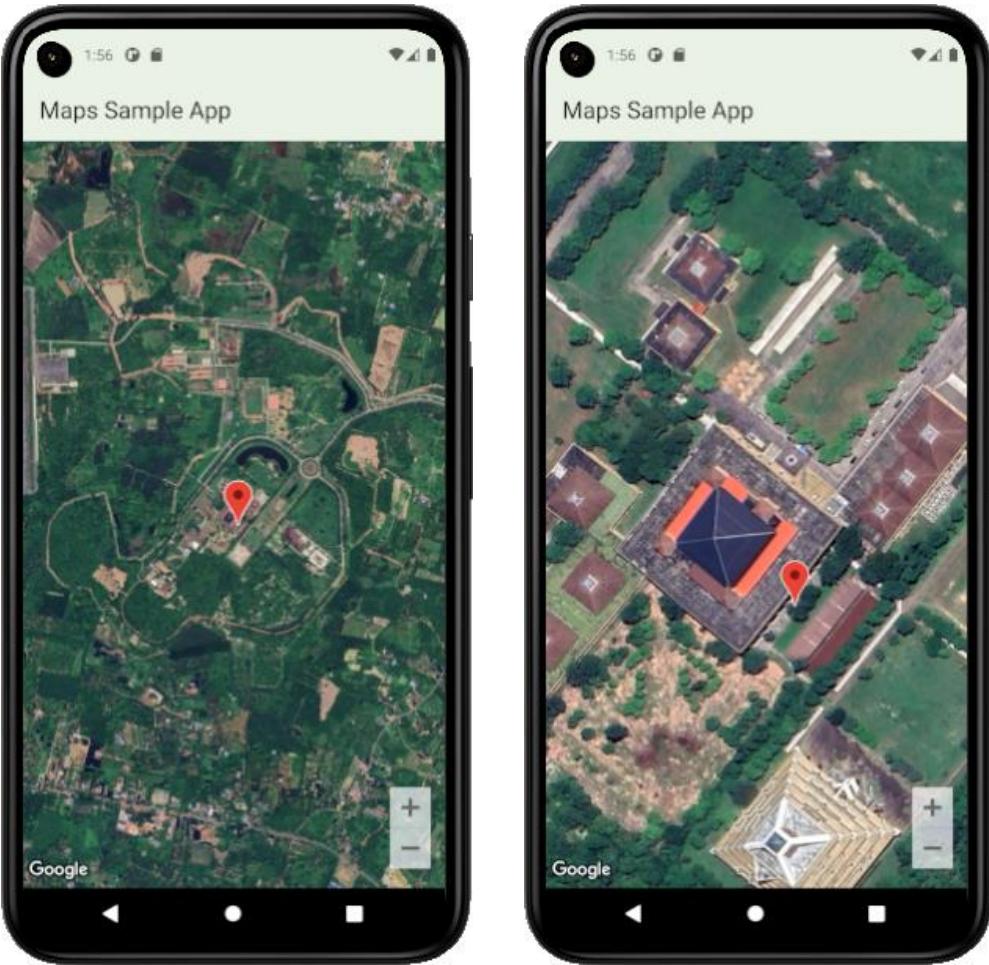
```
class _MyHomePageState extends State<MyHomePage> {  
    late GoogleMapController mapController;
```

Library จะมี Widget มาให้เช่นว่า GoogleMap() เพื่อแสดงแผนที่

เริ่มต้น GoogleMap จะต้อง initialCameraPosition เสมอ โดยกำหนดพิกัดจุดเริ่มต้นสถานที่จะให้ GoogleMap แสดงขึ้นมา

main.dart

```
1 import 'package:flutter/material.dart';
2 import 'package:google_maps_flutter/google_maps_flutter.dart';
3
4 void main() {
5   runApp(const MyApp());
6 }
7
8 class MyApp extends StatefulWidget {
9   const MyApp({super.key});
10
11   @override
12   State<MyApp> createState() => _MyAppState();
13 }
14
15 class _MyAppState extends State<MyApp> {
16   late GoogleMapController mapController;
17
18   final LatLng _center = const LatLng(14.1585968, 101.3457501);
19   final Set<Marker> _markers = Set<Marker>();
20
21   void _onMapCreated(GoogleMapController controller) {
22     mapController = controller;
23     _addMarker(_center, 'My Marker');
24   }
25
26   void _addMarker(LatLng position, String markerId) {
27     setState(() {
28       _markers.add(
29         Marker(
30           markerId: MarkerId(markerId),
31           position: position,
32           infoWindow: InfoWindow(title: markerId),
33         ),
34       );
35     });
36   }
37
38   @override
39   Widget build(BuildContext context) {
40     return MaterialApp(
41       debugShowCheckedModeBanner: false,
42       theme: ThemeData(useMaterial3: true, colorSchemeSeed: Colors.green[700]),
43       home: Scaffold(
44         appBar: AppBar(title: const Text('Maps Sample App'), elevation: 2),
45         body: GoogleMap(
46           onMapCreated: _onMapCreated,
47           initialCameraPosition: CameraPosition(target: _center, zoom: 16.0),
48           markers: _markers,
49           mapType: MapType.satellite,
50         ),
51       ),
52     );
53   }
54 }
55 }
```



** สามารถกำหนดระเบยการซูมแผนที่ ได้ตั้งแต่ 0 – 20

** สามารถรูปแบบการแสดงของแผนที่ จาก mapType ได้

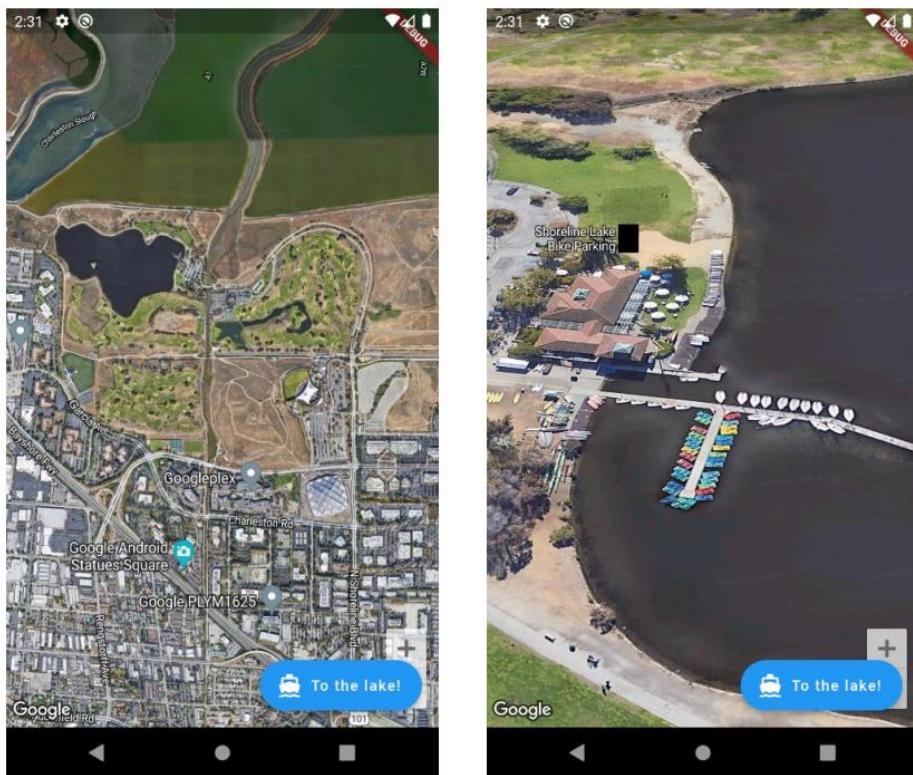
การกำหนด MapType สำหรับรูปแบบการแสดงผลแผนที่

```
45 |     body: GoogleMap
46 |     mapType: MapType.,
47 |     initialCameraPosi hybrid MapType
48 |     onMapCreated: (Go none
49 |       _controller.com normal
50 |       , satellite
51 |       // GoogleMap terrain
52 |       floatingActionButton values
```

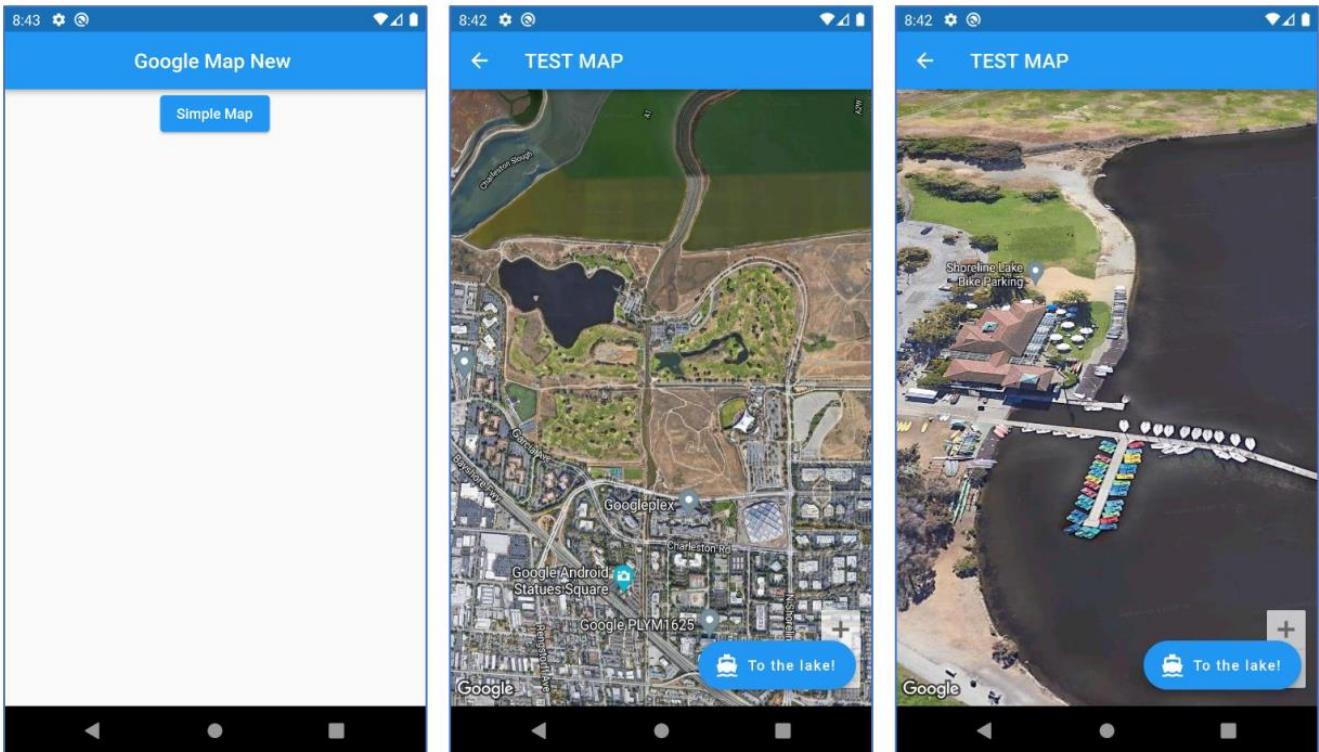
ปรับแก้ main.dart เพิ่มเติม แล้วทดสอบการแสดงผลการทำงานของแอปพลิเคชัน

```
1 import 'dart:async';
2
3 import 'package:flutter/material.dart';
4 import 'package:google_maps_flutter/google_maps_flutter.dart';
5
6 void main() {
7   runApp(MyApp());
8 }
9
10 class MyApp extends StatelessWidget {
11   //const MyApp({super.key});
12
13   @override
14   Widget build(BuildContext context) {
15     return MaterialApp(home: MapSample());
16   }
17 }
18
19 class MapSample extends StatefulWidget {
20   //const MapSample({super.key});
21
22   @override
23   State<MapSample> createState() => _MapSampleState();
24 }
25
26 class _MapSampleState extends State<MapSample> {
27   Completer<GoogleMapController> _controller = Completer();
28
29   static final CameraPosition _kGooglePlex = CameraPosition(
30     target: LatLng(37.42796133580664, -122.085749655962),
31     zoom: 14.0,
32   );
33
34   static final CameraPosition _kLake = CameraPosition(
35     bearing: 192.8334901395799,
36     target: LatLng(37.43296265331129, -122.088832357078792),
37     tilt: 59.44,
38     zoom: 19.0,
39   );
40
41   @override
42   Widget build(BuildContext context) {
43     return new Scaffold(
44       body: GoogleMap(
45         mapType: MapType.hybrid,
46         initialCameraPosition: _kGooglePlex,
47         onMapCreated: (GoogleMapController controller) {
48           _controller.complete(controller);
49         },
50       ),
51       floatingActionButton: FloatingActionButton.extended(
52         onPressed: _goToTheLake,
53         label: Text("Go to the Lake"),
54         icon: Icon(Icons.directions_boat),
55       ),
56     );
57   }
58
59   Future<void> _goToTheLake() async {
60     final GoogleMapController controller = await _controller.future;
61     controller.animateCamera(CameraUpdate.newCameraPosition(_kLake));
62   }
63 }
```

ทดสอบการทำงานของแอปพลิเคชัน



ปรับรูปแบบการแสดงผลของให้สามารถกลับไปที่หน้าแรกของ App ได้

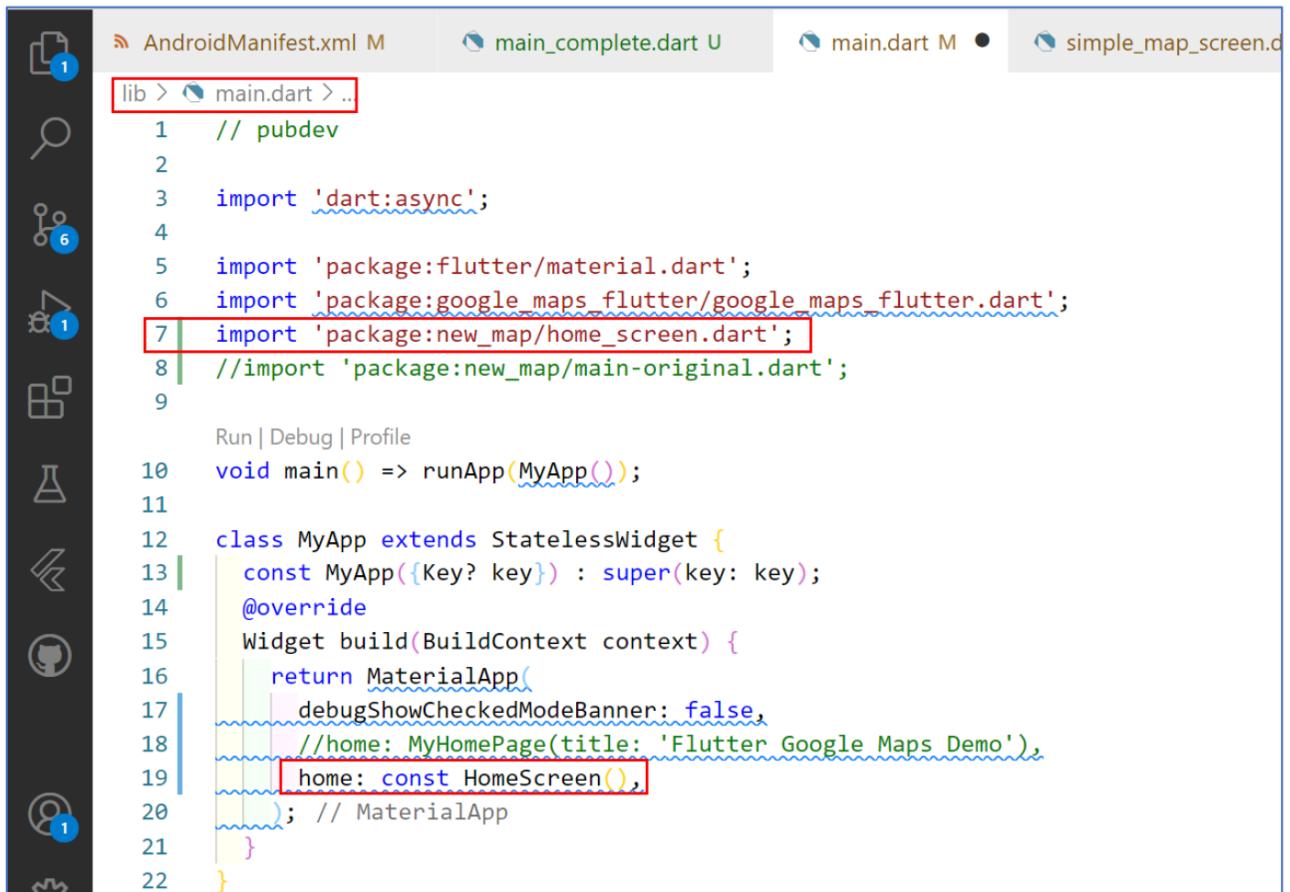


สร้างไฟล์ home_screen.dart ให้อยู่ใน folder lib ที่มี main.dart

```
lib > home_screen.dart > ..  
1 import 'package:flutter/cupertino.dart';  
2 import 'package:flutter/material.dart';  
3 import 'package:new_map/screens/simple_map_screen.dart';  
4  
5 class HomeScreen extends StatefulWidget {  
6     const HomeScreen({Key? key}) : super(key: key);  
7  
8     @override  
9     State<HomeScreen> createState() => _HomeScreenState();  
10 }  
11  
12 class _HomeScreenState extends State<HomeScreen> {  
13     @override  
14     Widget build(BuildContext context) {  
15         return Scaffold(  
16             appBar: AppBar(  
17                 title: const Text("Google Map New "),  
18                 centerTitle: true,  
19             ), // AppBar  
20             body: SizedBox(  
21                 width: MediaQuery.of(context).size.width,  
22                 child: Column(  
23                     children: [  
24                         ElevatedButton(  
25                             onPressed: () {  
26                                 Navigator.of(context)  
27                                     .push(MaterialPageRoute(builder: (BuildContext context) {  
28                                         return MapSample();  
29                                     })); // MaterialPageRoute  
30                             },  
31                             child: const Text("Simple Map")) // ElevatedButton  
32                         ],  
33                     ), // Column  
34                 ), // SizedBox  
35             ); // Scaffold  
36 }  
37 }
```

แก้ไขไฟล์ main.dart ให้ไปเรียกใช้ class ของ home_screen.dart

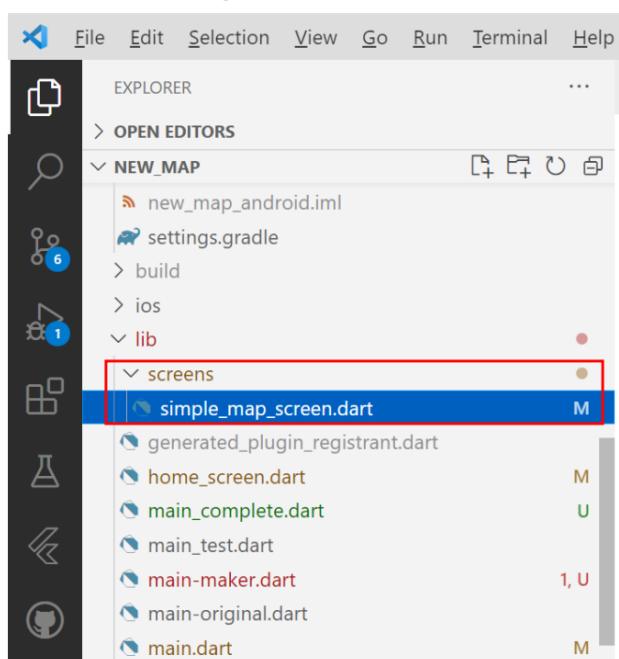
และสร้างการแสดง GUI ของ App ที่ไฟล์ simple_map_screen.dart ที่จะสร้างเพิ่มต่อไป



```
lib > main.dart > ...
1 // pubdev
2
3 import 'dart:async';
4
5 import 'package:flutter/material.dart';
6 import 'package:google_maps_flutter/google_maps_flutter.dart';
7 import 'package:new_map/home_screen.dart';
8 //import 'package:new_map/main-original.dart';
9
10 void main() => runApp(MyApp());
11
12 class MyApp extends StatelessWidget {
13   const MyApp({Key? key}) : super(key: key);
14   @override
15   Widget build(BuildContext context) {
16     return MaterialApp(
17       debugShowCheckedModeBanner: false,
18       //home: MyHomePage(title: 'Flutter Google Maps Demo'),
19       home: const HomeScreen(),
20     ); // MaterialApp
21   }
22 }
```

ทำการสร้าง folder screens ภายใต้ folder lib

จากนั้นสร้างไฟล์ simple_map_screen.dart ให้อยู่ใน folder screens

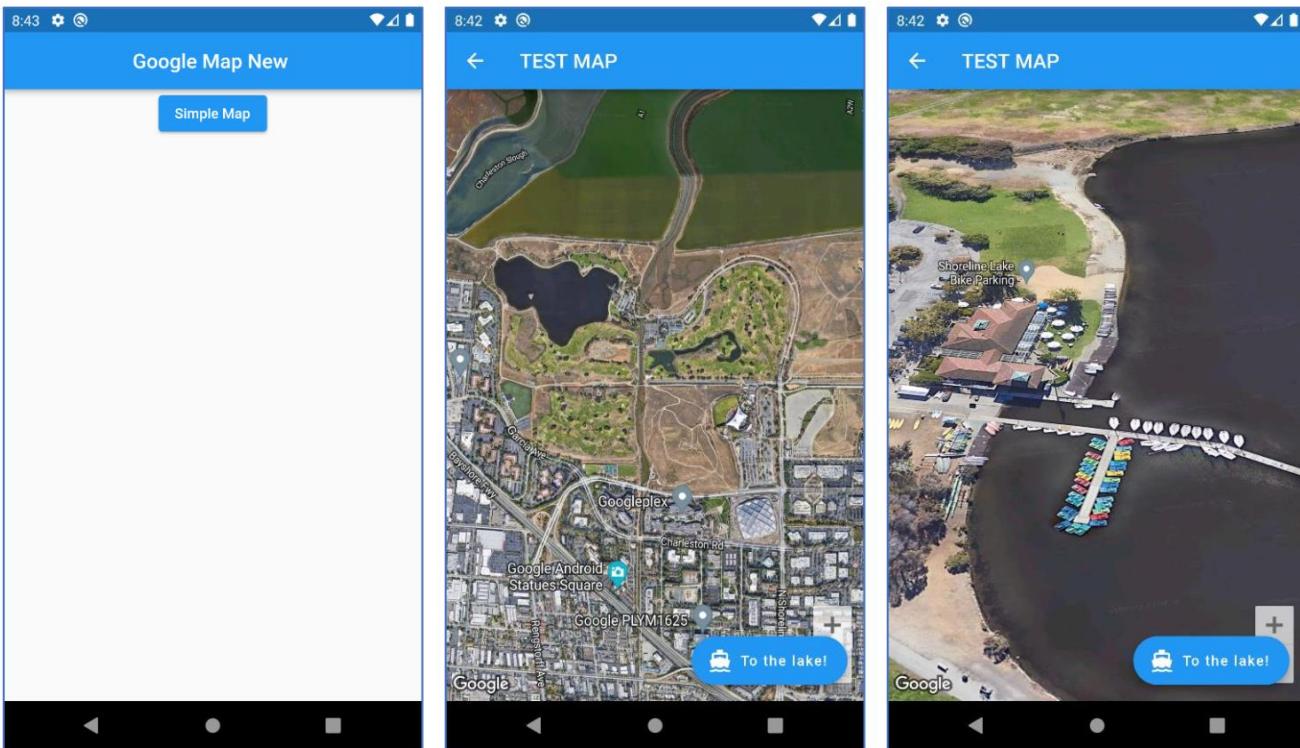


*** หน้า GUI ที่สร้างนี้ จะถูกเรียกใช้งาน เมื่อกดปุ่ม simple map ที่หน้าแรกของ APP

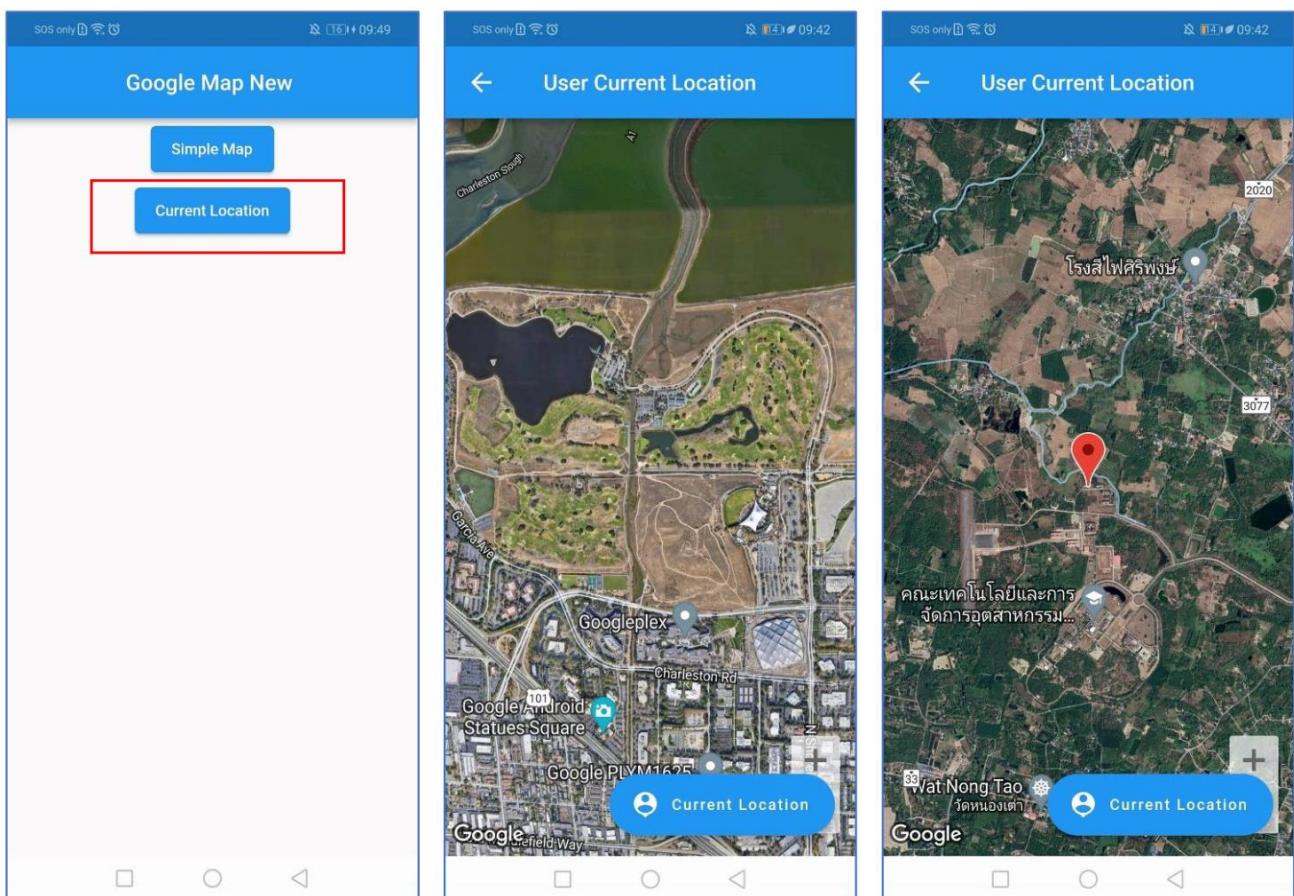
ไฟล์ simple_map_screen.dart

```
lib > screens > simple_map_screen.dart > MapSampleState
1 import 'dart:async';
2
3 import 'package:flutter/material.dart';
4 import 'package:google_maps_flutter/google_maps_flutter.dart';
5
6 class MapSample extends StatefulWidget {
7   @override
8   State<MapSample> createState() => MapSampleState();
9 }
10
11 class MapSampleState extends State<MapSample> {
12   Completer<GoogleMapController> _controller = Completer();
13
14   static final CameraPosition _kGooglePlex = CameraPosition(
15     target: LatLng(37.42796133580664, -122.085749655962),
16     zoom: 14.4746,
17   ); // CameraPosition
18
19   static final CameraPosition _kLake = CameraPosition(
20     bearing: 192.8334901395799,
21     target: LatLng(37.43296265331129, -122.08832357078792),
22     tilt: 59.440717697143555,
23     zoom: 19.151926040649414); // CameraPosition
24
25   @override
26   Widget build(BuildContext context) {
27     return new Scaffold(
28       appBar: AppBar(
29         title: Text("TEST MAP"),
30       ), // AppBar
31       body: GoogleMap(
32         mapType: MapType.hybrid,
33         initialCameraPosition: _kGooglePlex,
34         onMapCreated: (GoogleMapController controller) {
35           _controller.complete(controller);
36         },
37       ), // GoogleMap
38       floatingActionButton: FloatingActionButton.extended(
39         onPressed: _goToTheLake,
40         label: Text('To the lake!'),
41         icon: Icon(Icons.directions_boat),
42       ), // FloatingActionButton.extended
43     ); // Scaffold
44   }
45
46   Future<void> _goToTheLake() async {
47     final GoogleMapController controller = await _controller.future;
48     controller.animateCamera(CameraUpdate.newCameraPosition(_kLake));
49   }
50 }
```

ทดสอบการทำงานของ APP



เพิ่มการทำงานให้แสดง marker ที่พิกัดตำแหน่งปัจจุบันของอุปกรณ์

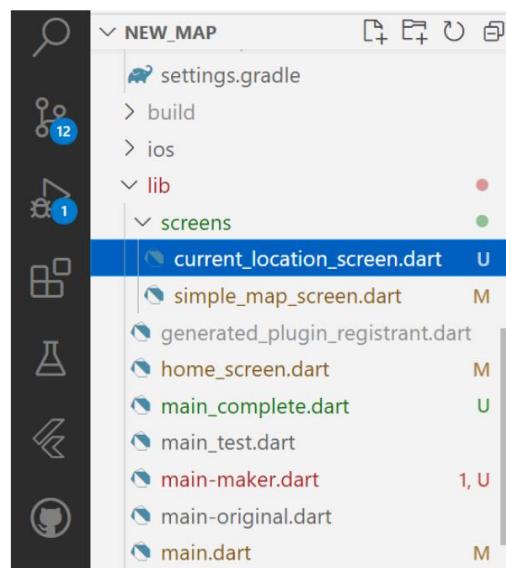


แก้ไข home_screen.dart ให้เพิ่มปุ่มสำหรับเรียกหน้า GUI สำหรับแสดงตำแหน่งปัจจุบัน

lib > home_screen.dart > _HomeScreenState > build

```
17     appBar: AppBar(
18       title: const Text("Google Map New "),
19       centerTitle: true,
20     ), // AppBar
21     body: SizedBox(
22       width: MediaQuery.of(context).size.width,
23       child: Column(
24         children: [
25           ElevatedButton(
26             onPressed: () {
27               Navigator.of(context)
28                 .push(MaterialPageRoute(builder: (BuildContext context) {
29                   return MapSample();
30                 })); // MaterialPageRoute
31             },
32             child: const Text("Simple Map")), // ElevatedButton
33           ElevatedButton(
34             onPressed: () {
35               Navigator.of(context)
36                 .push(MaterialPageRoute(builder: (BuildContext context) {
37                   return CurrentLocation();
38                 })); // MaterialPageRoute
39             },
40             child: const Text("Current Location")) // ElevatedButton
41         ],
42       ), // Column
43     ), // SizedBox
44   ); // Scaffold
45 }
46 }
```

สร้างไฟล์ current_location_screen.dart ใน folder screens



ไฟล์ current_location_screen.dart

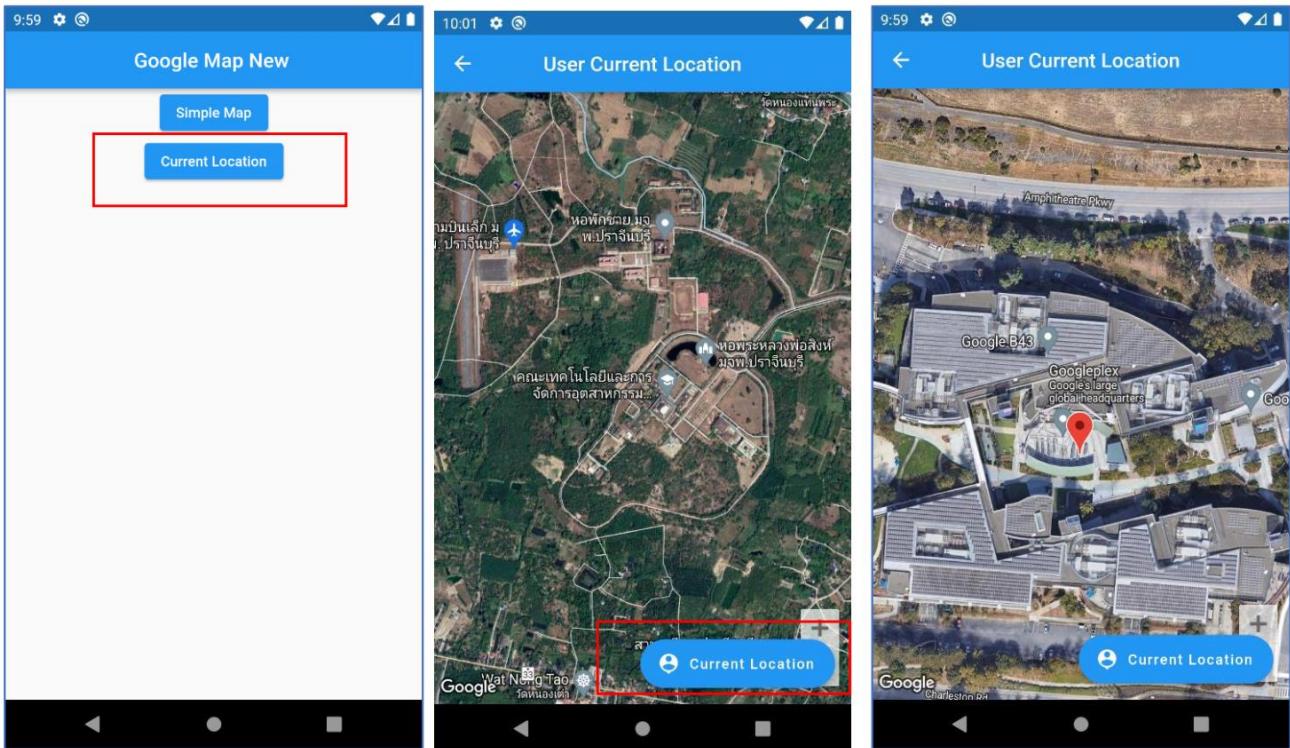
```
lib > screens > current_location_screen.dart > _CurrentLocationState > build
1 import 'package:flutter/material.dart';
2 import 'package:geolocator/geolocator.dart'; ←
3 import 'package:google_maps_flutter/google_maps_flutter.dart';
4
5 class CurrentLocation extends StatefulWidget {
6   const CurrentLocation({Key? key}) : super(key: key);
7
8   @override
9   State<CurrentLocation> createState() => _CurrentLocationState();
10 }
11
12 class _CurrentLocationState extends State<CurrentLocation> {
13   late GoogleMapController googleMapController;
14
15   static const CameraPosition _initialCameraPosition = CameraPosition(
16     target: LatLng(37.42796133580664, -122.085749655962), zoom: 14.4746); // CameraPosition
17
18   Set<Marker> markers = {};
19   @override
20   Widget build(BuildContext context) {
21     return Scaffold(
22       appBar: AppBar(
23         title: Text("User Current Location"),
24         centerTitle: true,
25       ), // AppBar
26       body: GoogleMap(
27         markers: markers,
28         mapType: MapType.hybrid,
29         initialCameraPosition: _initialCameraPosition,
30         onMapCreated: (GoogleMapController controller) {
31           googleMapController = controller;
32         },
33       ), // GoogleMap
34       floatingActionButton: FloatingActionButton.extended(
35         onPressed: () async {
36           Position position = await _determinePosition();
37           googleMapController.animateCamera(CameraUpdate.newCameraPosition(
38             CameraPosition(
39               target: LatLng(position.latitude, position.longitude),
40               zoom: 14)); // CameraPosition
41           markers.clear();
42           markers.add(Marker(
43             markerId: MarkerId("Current Location"),
44             position: LatLng(position.latitude, position.longitude))); // Marker
45           setState(() {});
46         },
47         label: const Text("Current Location"),
48         icon: const Icon(Icons.location_history),
49       ), // FloatingActionButton.extended
50     ); // Scaffold
51   }
52
53 Future<Position> _determinePosition() async {
54   bool serviceEnable;
55   LocationPermission permission;
56
57   serviceEnable = await Geolocator.isLocationServiceEnabled();
58   if (!serviceEnable) {
59     return Future.error("Location Service Disable");
60   }
61
62   permission = await Geolocator.checkPermission();
63   if (permission == LocationPermission.denied) {
64     permission = await Geolocator.requestPermission();
65
66     if (permission == LocationPermission.denied) {
67       return Future.error("Location Permission Denied");
68     }
69   }
70
71   if (permission == LocationPermission.deniedForever) {
72     return Future.error("Location Permission are permanently denied");
73   }
74
75   Position position = await Geolocator.getCurrentPosition();
76   return position;
77 }
78 }
```

```
dependencies:
  flutter:
    sdk: flutter

  # The following adds the Cupertino Icons font to your application.
  # For more details, see https://flutter.dev/docs/development/ui/assets-and-images#using-cupertino-icons
  cupertino_icons: ^1.0.8
  google_maps_flutter: ^2.10.0
  geolocator: ^13.0.2
```

** ติดตั้ง geolocator:^13.0.2

ทดสอบด้วย Emulator เมื่อกดปุ่ม Current Location จะแสดงตำแหน่งที่ Googleplex



ให้ทำการ Build apk แล้วติดตั้งบนอุปกรณ์จริง เพื่อทดสอบการทำงาน
เมื่อกดปุ่ม Current Location จะแสดงตำแหน่งของอุปกรณ์

