Mobile Development :

12 : Building the Beta Version : Part 3

Firebase Services, Google Maps

Professor Imed Bouchrika

National School of Artificial Intelligence
imed.bouchrika@ensia.edu.dz

Outline :

- ***Firebase Services***
  - ***Introduction, Setup and Configuration***
  - ***Firebase Cloud Storage***
  - ***Firebase Messaging***
  - ***Firebase Firestore***
  - ***Firebase Remote Config***

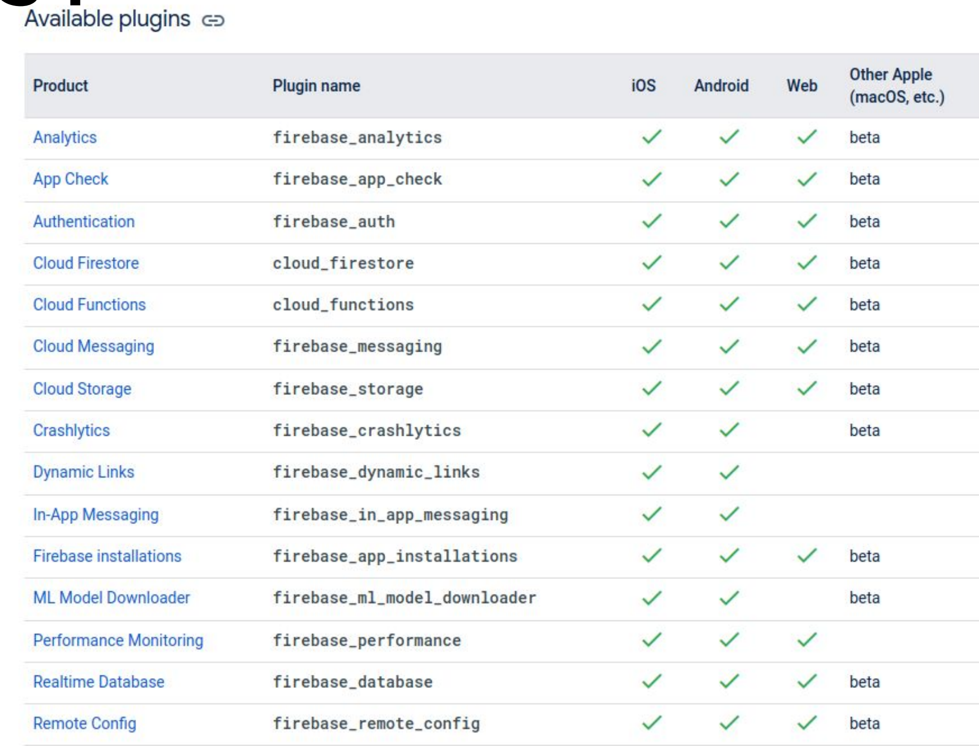
Firebase Services : Introduction

- **Firebase is :**
  - “a Backend-as-a-Service (BaaS) app development platform that provides hosted backend services such as a realtime database, cloud storage, authentication, crash reporting, machine learning, remote configuration, and hosting for your static files.“

https://docs.flutter.dev/data-and-backend/firebase?

Firebase Services : Introduction

- **Services Provided by Firebase :**
  - Messaging
  - Remote Config
  - Database (NoSQL)
  - File Storage
  - Authentication
  - Machine Learning
  - Analytics
  - Functions
  -



Available plugins

Product	Plugin name	iOS	Android	Web	Other Apple (macOS, etc.)
Analytics	`firebase_analytics`	✓	✓	✓	beta
App Check	`firebase_app_check`	✓	✓	✓	beta
Authentication	`firebase_auth`	✓	✓	✓	beta
Cloud Firestore	`cloud_firestore`	✓	✓	✓	beta
Cloud Functions	`cloud_functions`	✓	✓	✓	beta
Cloud Messaging	`firebase_messaging`	✓	✓	✓	beta
Cloud Storage	`firebase_storage`	✓	✓	✓	beta
Crashlytics	`firebase_crashlytics`	✓	✓		beta
Dynamic Links	`firebase_dynamic_links`	✓	✓		
In-App Messaging	`firebase_in_app_messaging`	✓	✓		
Firebase installations	`firebase_app_installations`	✓	✓	✓	beta
ML Model Downloader	`firebase_ml_model_downloader`	✓	✓		beta
Performance Monitoring	`firebase_performance`	✓	✓	✓	
Realtime Database	`firebase_database`	✓	✓	✓	beta
Remote Config	`firebase_remote_config`	✓	✓	✓	beta

List of all firebase plugins for flutter : https://firebase.google.com/docs/flutter/setup?platform=ios#available-plugins

Firebase Services : Introduction

- **Steps to get Started**
  - Install the Firebase CLI :
    - https://firebase.google.com/docs/cli#install-cli-mac-linux
  - Link your Firebase Account
    - **firebase login**
  - Follow all instructions at:

    https://firebase.google.com/docs/flutter/setup?platform=ios

Section 1

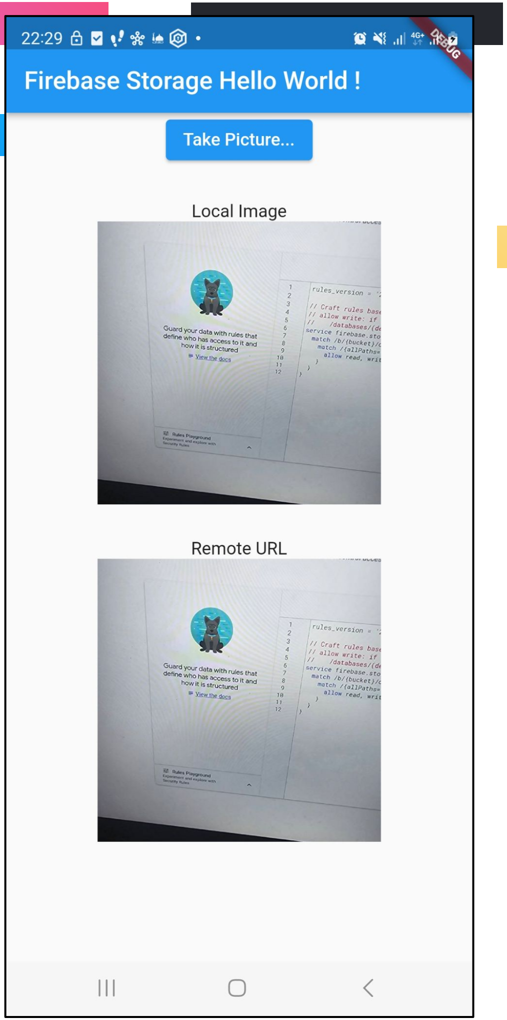
Cloud Storage

Firebase Services : Cloud Storage

- **Firebase Cloud Storage**
  - “built on fast and secure Google Cloud infrastructure for app developers who need to store and serve user-generated content, such as images or videos.”
  - Plugins to install
    - flutter pub add firebase_core
    - flutter pub add firebase_storage

Firebase Services : Cloud Storage

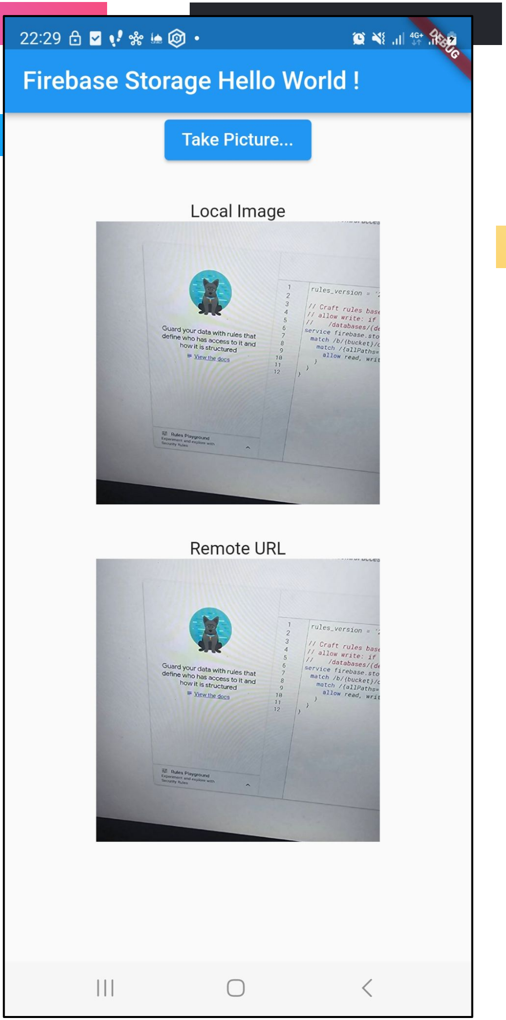
- **Firebase Cloud Storage : MVP (Hello World !)**
  - MVP :
    - Simple Image Picker
    - Take and show a photo stored locally
    - Upload **immediately** to Firebase
    - Get a Remote URL for the image
    - Show it on the UI



Firebase Services : Cloud Storage

main.dart

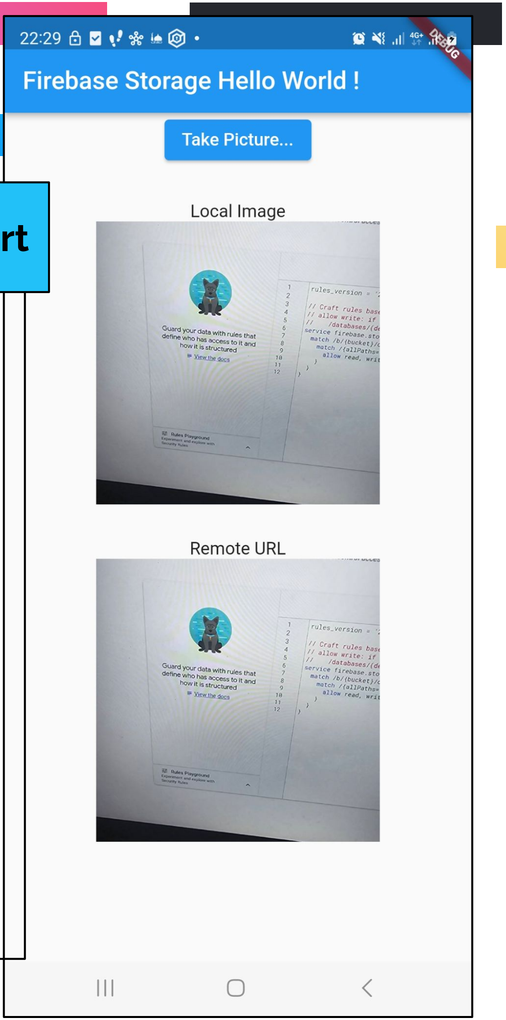
```
import 'firebase_options.dart';

final storage = FirebaseStorage.instance;
final storageRef = FirebaseStorage.instance.ref();

Future<bool> my_init_app() async {
 await Firebase.initializeApp(
   options: DefaultFirebaseOptions.currentPlatform,
 );
 return true;
}

void main() async {
 WidgetsFlutterBinding.ensureInitialized();
 await my init app();
 runApp(const MainApp());
}
```



Firebase Services : Cloud Storage

screens/home.dart

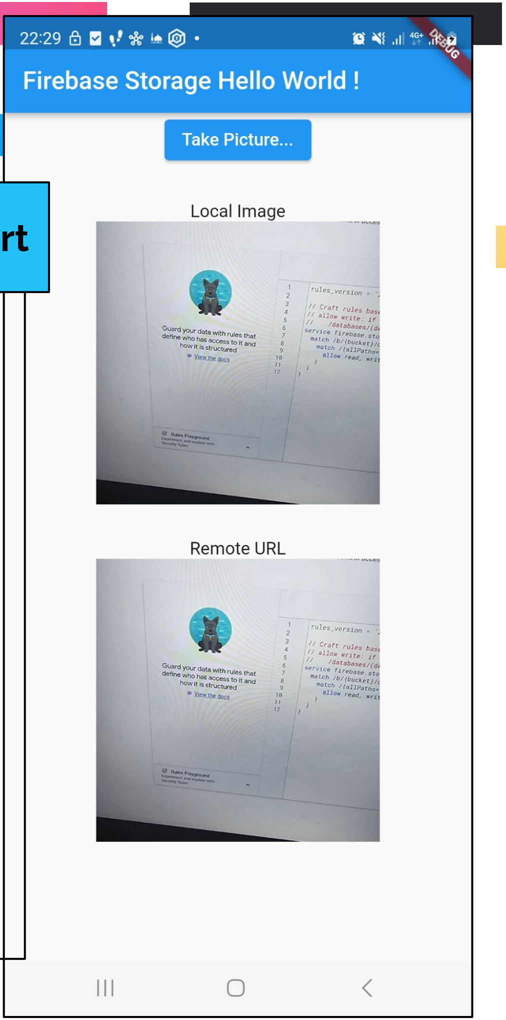
```
ElevatedButton(
    onPressed: () async {
      String? path = await onImageButtonPressed(
        ImageSource.camera,
        context,
      );
      if (path != null) {
        local path = path;
        setState(() {});

        upload_to_firebase_storage();
      }
    },
    child: Text("Take Picture...")),
```



Firebase Services : Cloud Storage

screens/home.dart

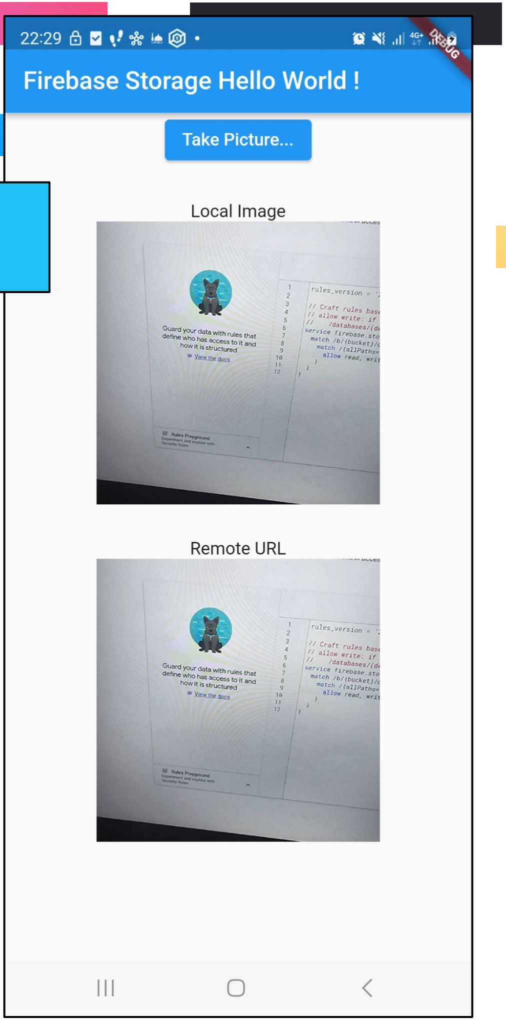
```
Future<bool> upload_to_firebase_storage() async {
  try {
    String filename = 'abc.jpg';
    final spaceRef = storageRef.child("myimages/$filename");
    await spaceRef.putFile(File(local_path));
    final imageUrl = remote_url = await spaceRef.getDownloadURL();
    setState(() {});
  } catch (e) {
    print("Exception $e");
  }
  return true;
}
```

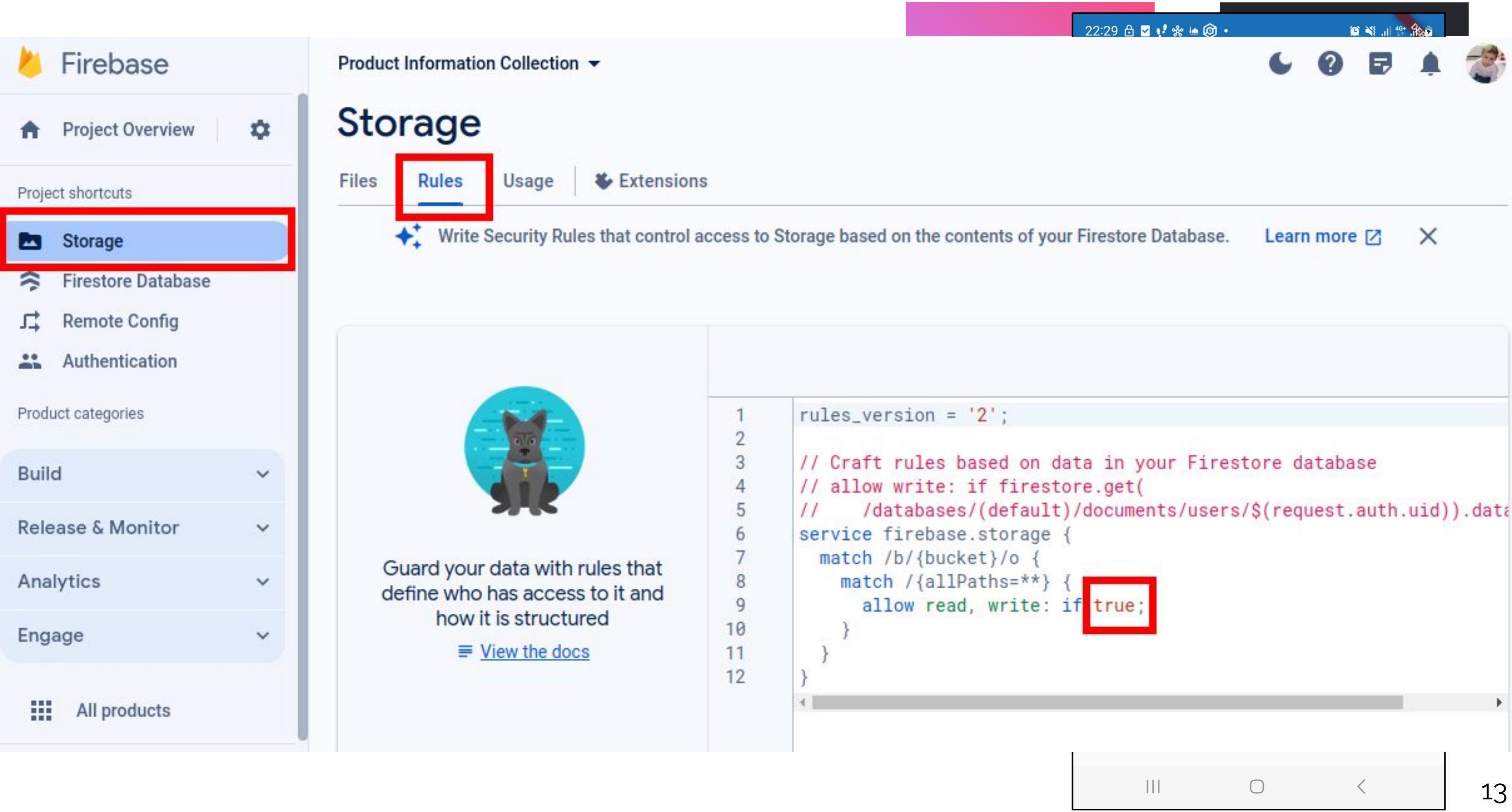


Firebase Services : Cloud Storage

IMPORTANT ..

- **Firebase Cloud Storage**
  - Login to Firebase Console and Initialize the bucket
  - Grant the security rules to write, read to **true**



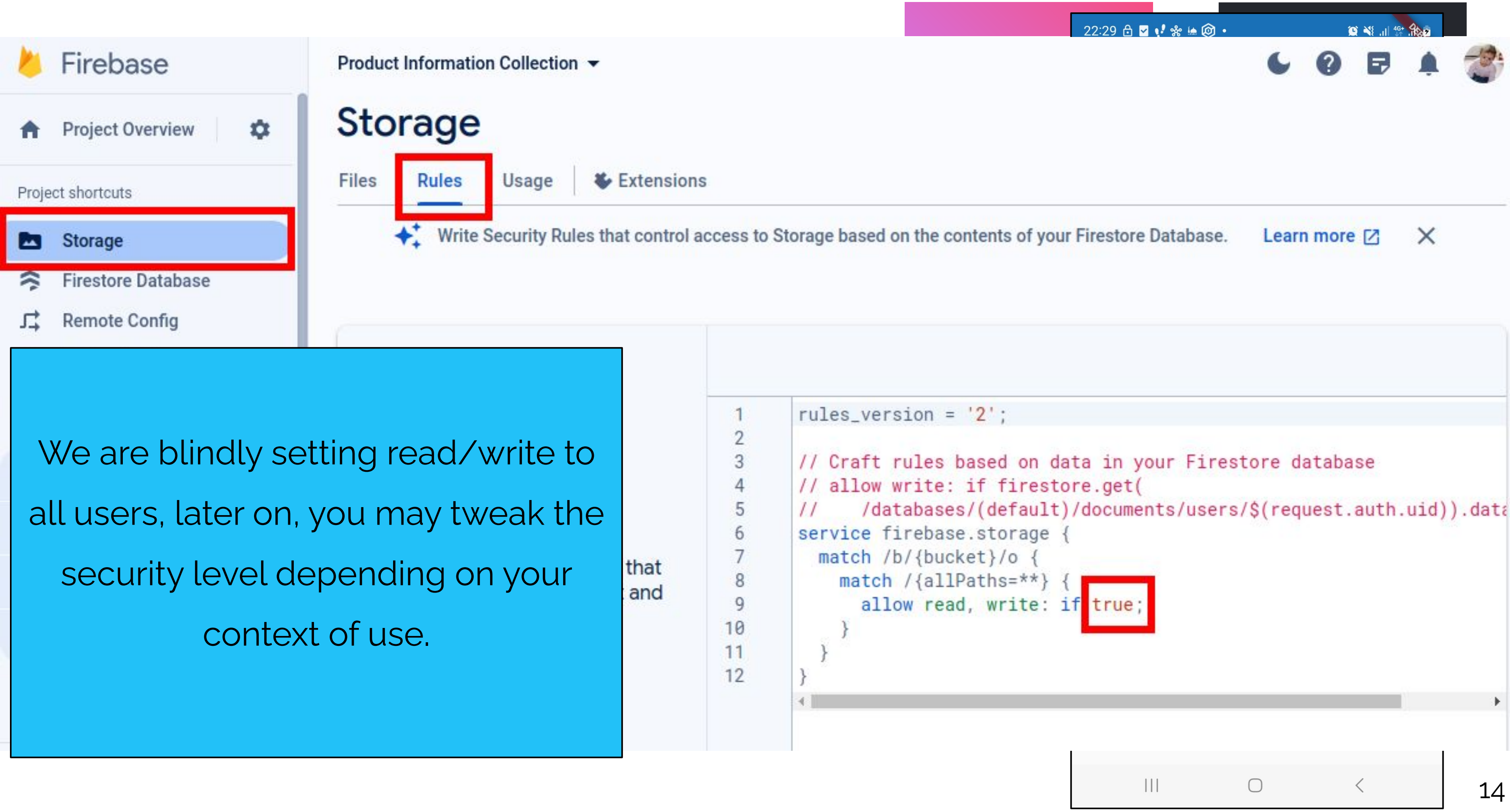


Storage

Write Security Rules that control access to Storage based on the contents of your Firestore Database.

Guard your data with rules that define who has access to it and how it is structured

```
rules_version = '2';

// Craft rules based on data in your Firestore database
// allow write: if firestore.get(
//    /databases/(default)/documents/users/$(request.auth.uid)).data
service firebase.storage {
  match /b/{bucket}/o {
    match /{allPaths=**} {
      allow read, write: if true;
    }
  }
}
```



Storage

Write Security Rules that control access to Storage based on the contents of your Firestore Database.

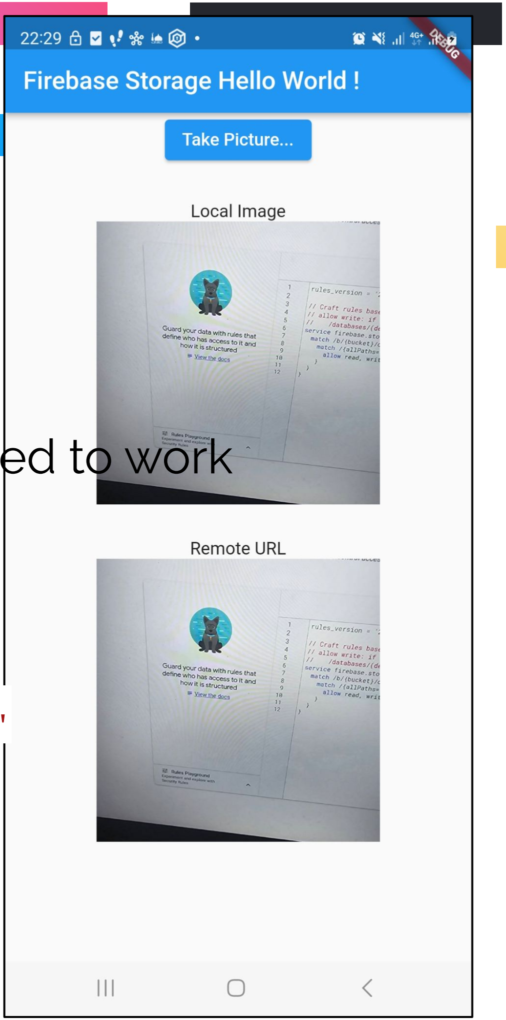
```
rules_version = '2';

// Craft rules based on data in your Firestore database
// allow write: if firestore.get(
//    /databases/(default)/documents/users/$(request.auth.uid)).data
service firebase.storage {
  match /b/{bucket}/o {
    match /{allPaths=**} {
      allow read, write: if true;
    }
  }
}
```

that
and

We are blindly setting read/write to all users, later on, you may tweak the security level depending on your context of use.

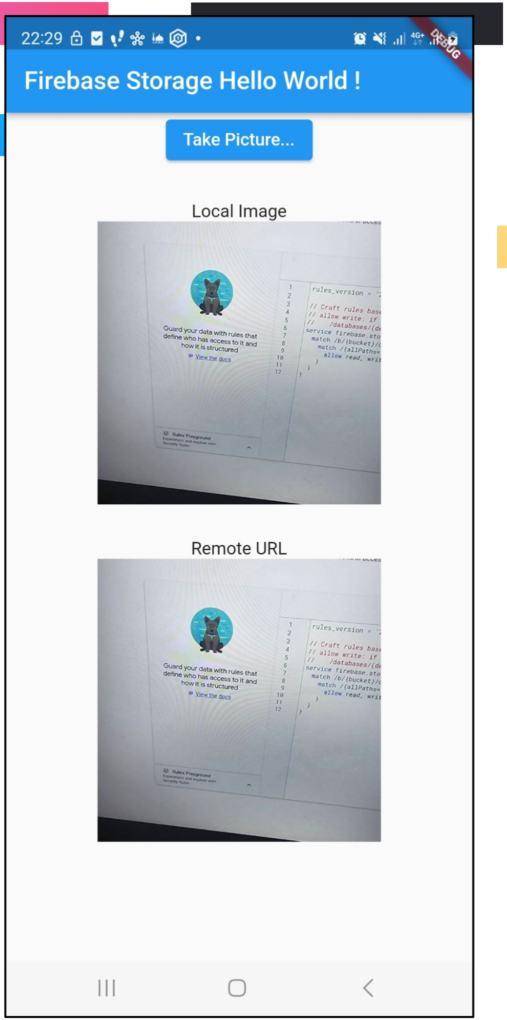
Firebase Services : Cloud Storage

- **Firebase Cloud Storage**
  - Errors & Failure when building ? Try (not guaranteed to work
    - Update the Google Play Services
      - android/build.gradle
        - `classpath 'com.google.gms:google-services:4.4.0'`
    - Increase the minSdk
      - android/app/src/build.gradle
        - `minSdkVersion 19`



Firebase Services : Cloud Storage

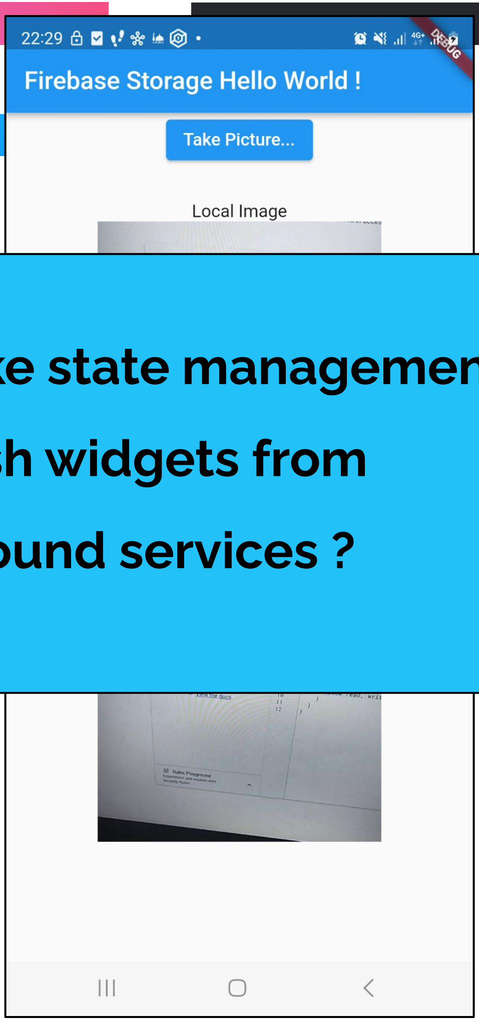
- **Warning**
  - We don't upload directly to backed
  - This was an easy hello world to show you how to upload.
  - In real cases, use the principles taught in the previous lecture:
    - *Store upload tasks into a local DB*
    - *Upload using a Cron Job*
    - *Use State Management when possible*



Firebase Services : Cloud Storage

- **Warning**
  - We don't upload directly to backed
  - This was an easy hello world to show yo
  - In real cases, use the principles taught i lecture:
    - *Store upload tasks into a local DB*
    - *Upload using a Cron Job*
    - *Use State Management when possible*

Can you invoke state management to refresh widgets from Background services ?

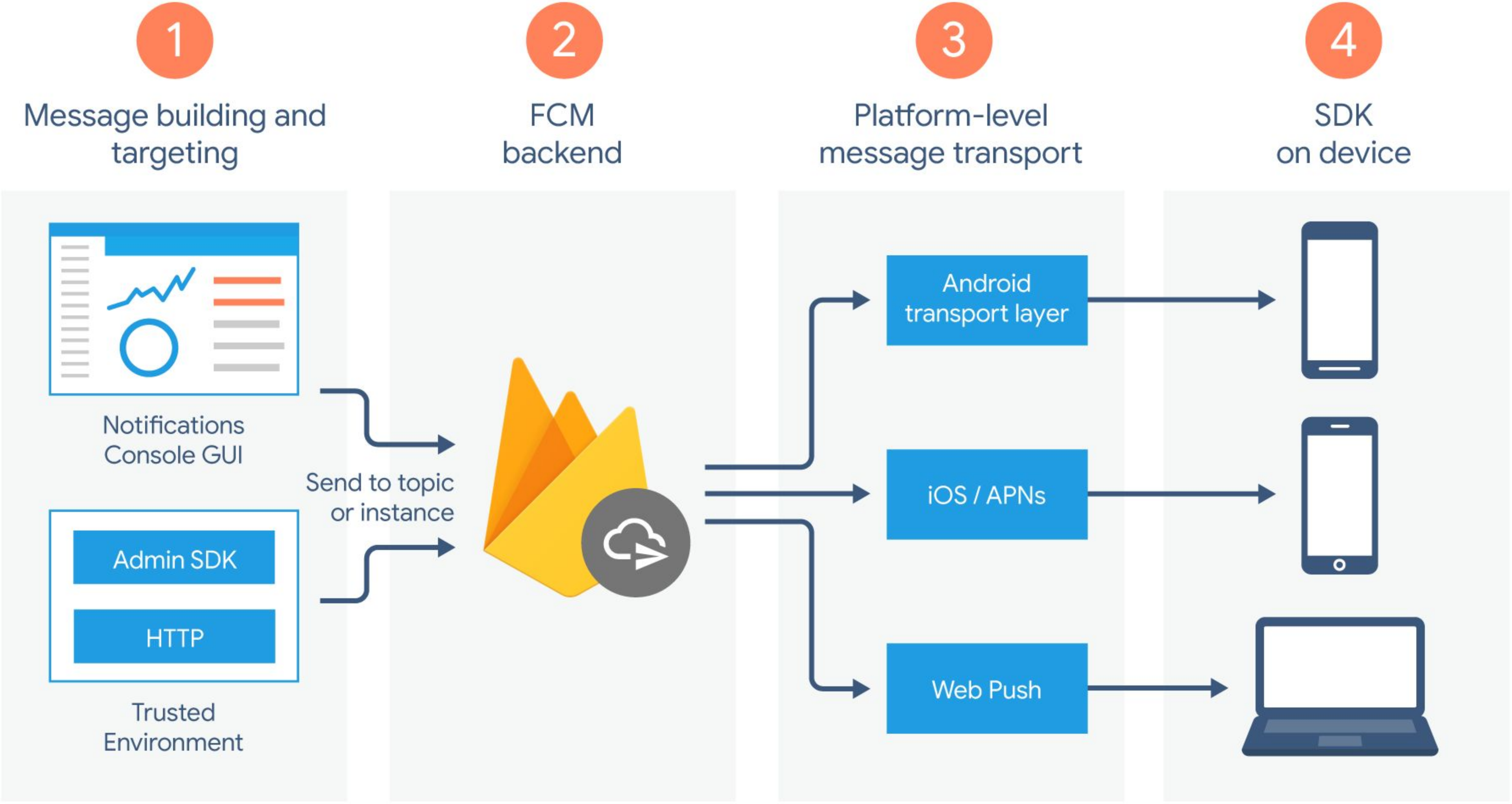


Section 2

Firebase Cloud Messaging

Firebase Services : Messaging

- **Firebase Messaging :**
  - Used either to send “almost” immediately **from remote systems**:
    - Notification Messages
    - In-App Messages (Shown inside the app whilst the user uses the app)
  - You can send to :
    - A single user
    - Group of users subscribed to a specific group.
  - The sending is **usually** triggered by a **backend server** which communicates with Firebase Servers



1 Message building and targeting

2 FCM backend

3 Platform-level message transport

4 SDK on device

Notifications Console GUI

Admin SDK

HTTP

Trusted Environment

Send to topic or instance

Android transport layer

iOS / APNs

Web Push

Firebase Services : Messaging

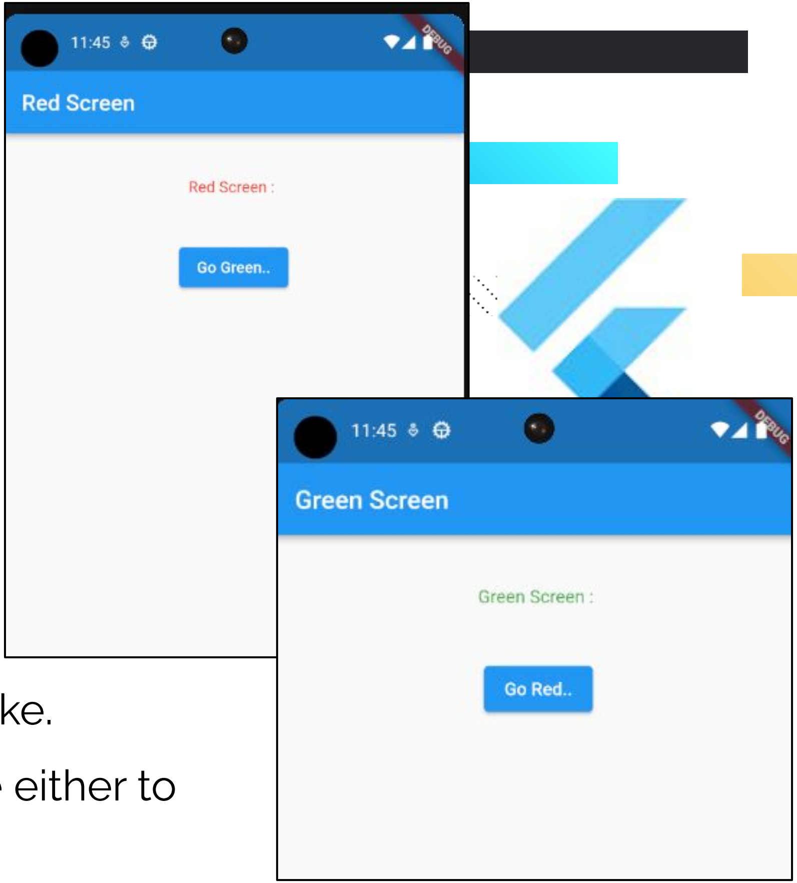
- **Firebase Messaging : Overview for a Hello World !**
    - **On Mobile App :**
        - Init the Firebase Messaging Instance
        - Depending on your App:
            - Get the Token for the User and send it to your Backend OR
            - Subscribe the User to a specific topic/channel
        - Implement Event Listeners when a message arrives
            - You may even take the user to a specific screen when clicking on the message

Firebase Services : Messaging

- **Firebase Messaging : Overview for a Hello World !**
  - **On Backend Server**
    - Implement UI/Service to send messages either to specific user or channel
    - You may provide an API Endpoint to get User Tokens

Firebase Services : Messaging

- **Firebase Messaging : Simple MVP**
  - Apps with two Screens :
    - Red
    - Green
  - Users can choose whichever screen they like.
  - Messages would be sent from time to time either to
    - All users
    - Users of specific color (Red or Green)

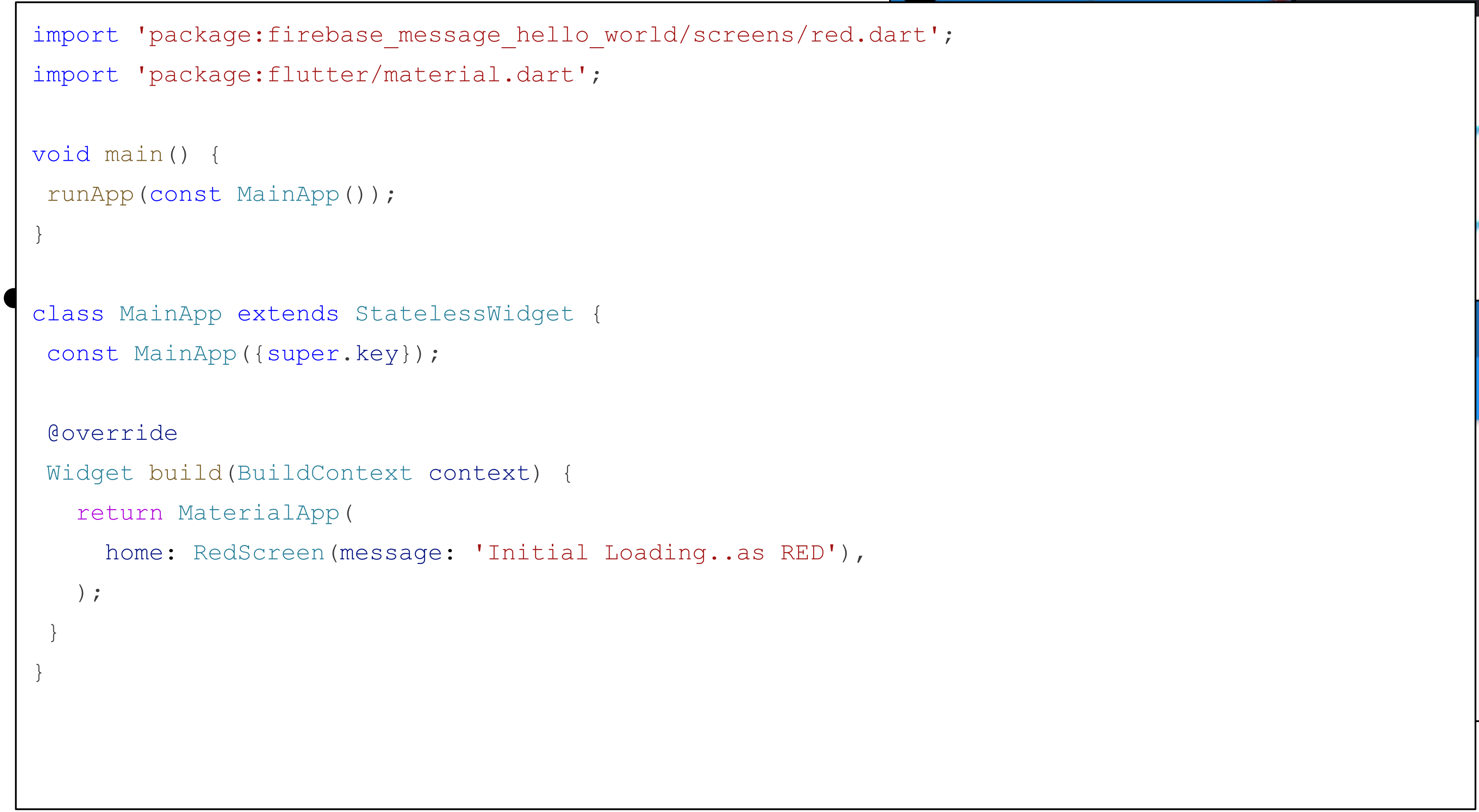


Red Screen

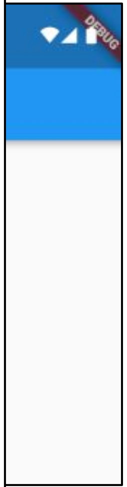
Red Screen :

Green Screen

Green Screen :



```dart
import 'package:firebase_message_hello_world/screens/red.dart';
import 'package:flutter/material.dart';

void main() {
 runApp(const MainApp());
}

class MainApp extends StatelessWidget {
 const MainApp({super.key});

 @override
 Widget build(BuildContext context) {
   return MaterialApp(
     home: RedScreen(message: 'Initial Loading..as RED'),
   );
 }
}
```

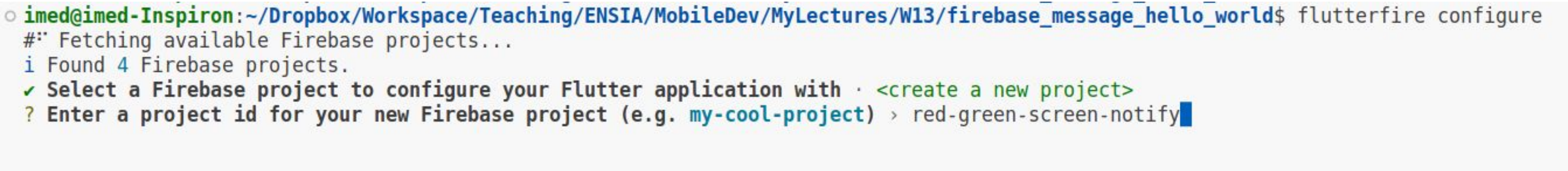


```
import 'package:firebase_message_hello_world/screens/green.dart;'
import 'package:flutter/material.dart';
class RedScreen extends StatefulWidget {
 String message;
 RedScreen({super.key, this.message = ''});
 @override
 State<RedScreen> createState() => _RedScreenState();
}
class _RedScreenState extends State<RedScreen> {
 @override
 Widget build(BuildContext context) {
   return Scaffold(
     appBar: AppBar(title: Text("Red Screen")),
     body: Center(
         child: Column(
       children: [
         SizedBox(height: 40,),
         Text("Red Screen : ${widget.message}",
             style: TextStyle(color: Colors.red)),
         SizedBox(height: 40,),
         ElevatedButton(
             onPressed: () {
               Navigator.of(context).pop();
               Navigator.of(context).push(
                   MaterialPageRoute(builder: (context) => GreenScreen()));
             },
             child: Text("Go Green.."))
       ],
     )),
   );
 }
}
```

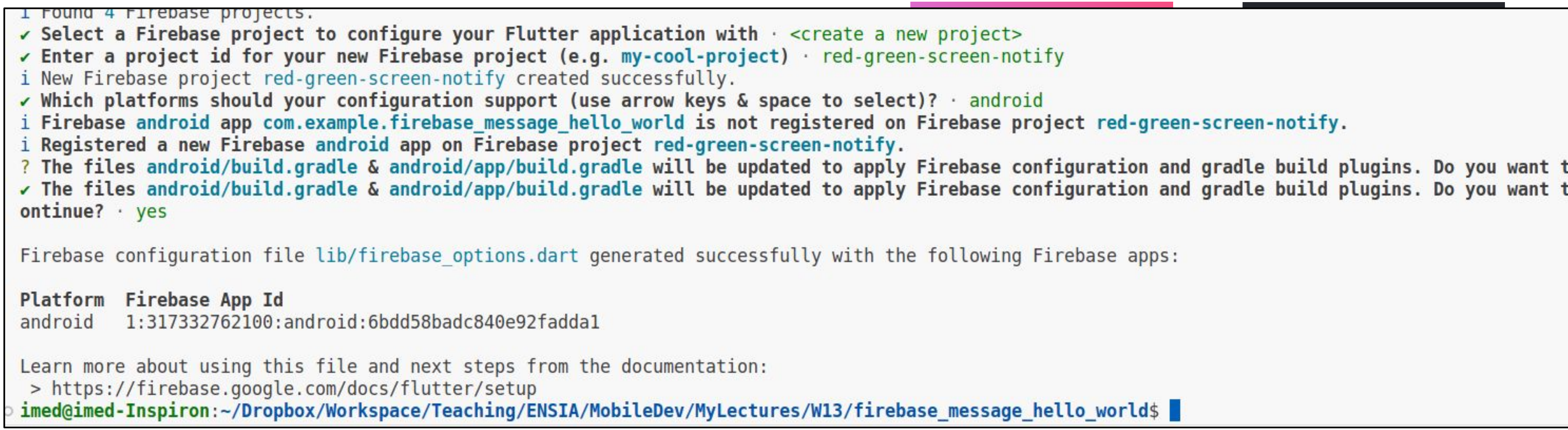
Firebase Services : Messaging

- **Firebase Messaging : Simple Steps for a Hello World !**
  - **Mobile App :**
    - Make sure that you have installed the firebase cli tools
    - Within you your project terminal run:
      - **flutterfire configure**
      - Either choose an existing project or create a new one

01



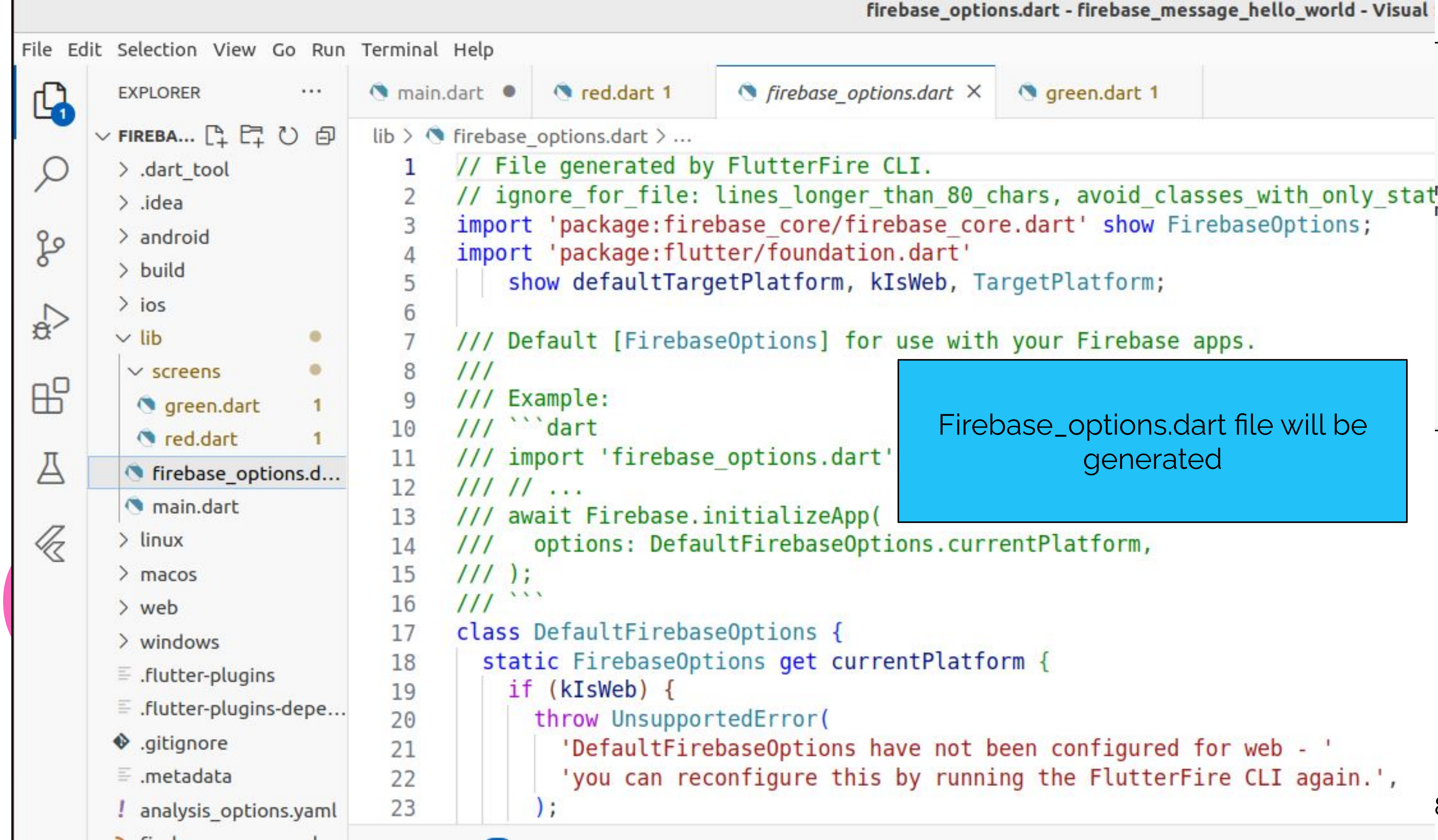
```
○ imed@imed-Inspiron:~/Dropbox/Workspace/Teaching/ENSIA/MobileDev/MyLectures/W13/firebase_message_hello_world$ flutterfire configure
  #⠋ Fetching available Firebase projects...
  i Found 4 Firebase projects.
  ✔ Select a Firebase project to configure your Flutter application with · <create a new project>
  ? Enter a project id for your new Firebase project (e.g. my-cool-project) › red-green-screen-notify
```



```
i Found 4 Firebase projects.
✔ Select a Firebase project to configure your Flutter application with · <create a new project>
✔ Enter a project id for your new Firebase project (e.g. my-cool-project) · red-green-screen-notify
i New Firebase project red-green-screen-notify created successfully.
✔ Which platforms should your configuration support (use arrow keys & space to select)? · android
i Firebase android app com.example.firebase_message_hello_world is not registered on Firebase project red-green-screen-notify.
i Registered a new Firebase android app on Firebase project red-green-screen-notify.
? The files android/build.gradle & android/app/build.gradle will be updated to apply Firebase configuration and gradle build plugins. Do you want t
✔ The files android/build.gradle & android/app/build.gradle will be updated to apply Firebase configuration and gradle build plugins. Do you want t
ontinue? · yes

Firebase configuration file lib/firebase_options.dart generated successfully with the following Firebase apps:

Platform  Firebase App Id
android   1:317332762100:android:6bdd58badc840e92fadda1

Learn more about using this file and next steps from the documentation:
 > https://firebase.google.com/docs/flutter/setup
imed@imed-Inspiron:~/Dropbox/Workspace/Teaching/ENSIA/MobileDev/MyLectures/W13/firebase_message_hello_world$
```

01

- Make sure that you have installed the firebase cli tools
- Within you your project terminal run:
  - **flutterfire configure**
  - Either choose an existing project or create a new one
  - You need to choose the platform you want to deploy the app (Android only ?)



```dart
// File generated by FlutterFire CLI.
// ignore_for_file: lines_longer_than_80_chars, avoid_classes_with_only_stat
import 'package:firebase_core/firebase_core.dart' show FirebaseOptions;
import 'package:flutter/foundation.dart'
    show defaultTargetPlatform, kIsWeb, TargetPlatform;

/// Default [FirebaseOptions] for use with your Firebase apps.
///
/// Example:
/// ```dart
/// import 'firebase_options.dart'
/// // ...
/// await Firebase.initializeApp(
///   options: DefaultFirebaseOptions.currentPlatform,
/// );
/// ```
class DefaultFirebaseOptions {
  static FirebaseOptions get currentPlatform {
    if (kIsWeb) {
      throw UnsupportedError(
        'DefaultFirebaseOptions have not been configured for web - '
        'you can reconfigure this by running the FlutterFire CLI again.',
      );
```

Firebase_options.dart file will be generated

Firebase Services : Messaging

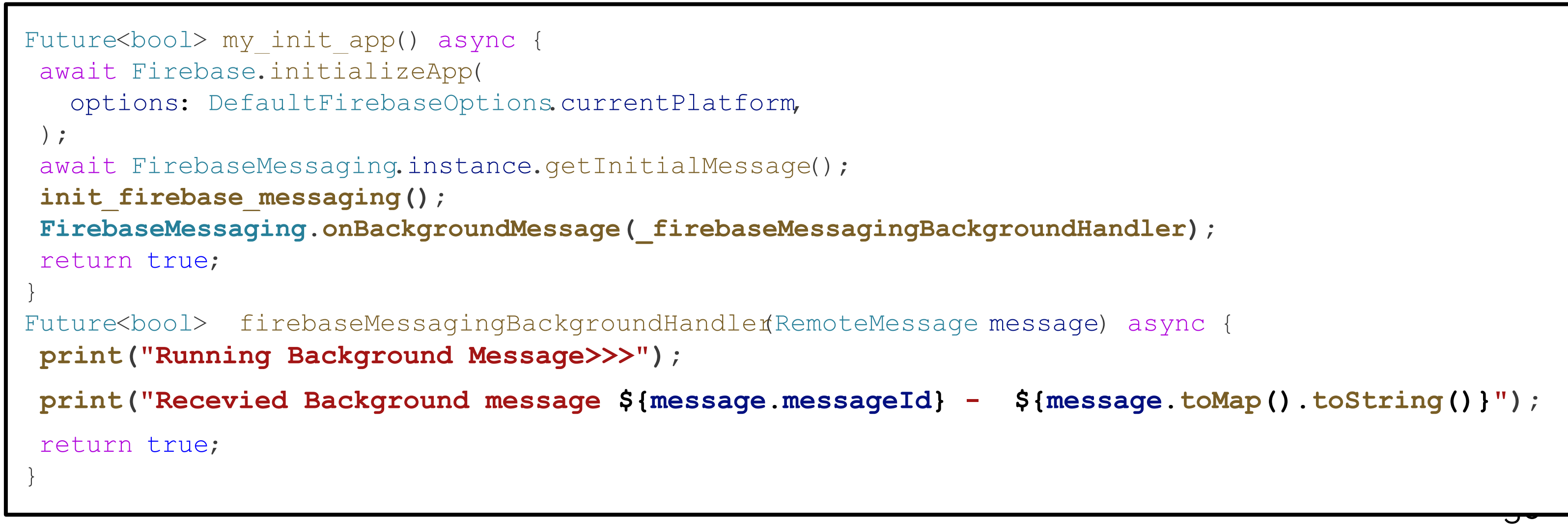
- **Firebase Messaging : Simple Steps for a Hello World !**
  - **Mobile App :**
    - Install the firebase plugins :
      - **flutter pub add firebase_core**
      - **flutter pub add firebase_messaging**
    - In addition to another plugin for showing messages:
      - **flutter pub add flutter_local_notifications**

02

Firebase Services : Messaging

- **Firebase Messaging : Simple Steps for a Hello World !**
  - ○ **Mobile App :**

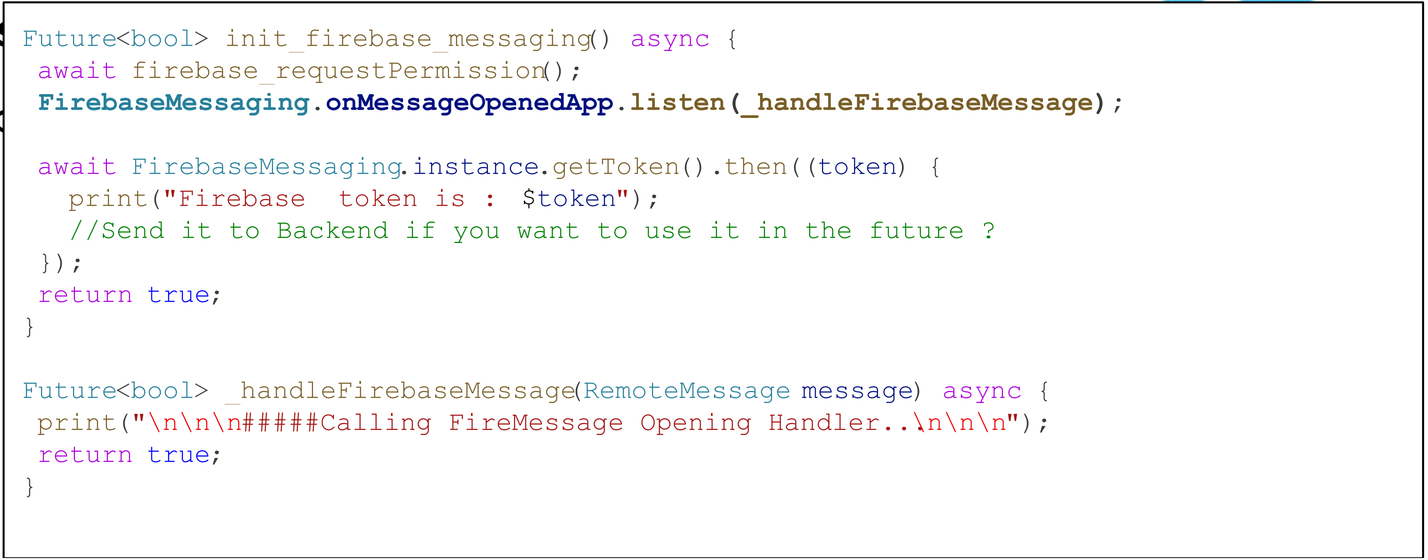
03



```
Future<bool> my_init_app() async {
 await Firebase.initializeApp(
   options: DefaultFirebaseOptions.currentPlatform,
 );
 await FirebaseMessaging.instance.getInitialMessage();
 init_firebase_messaging();
 FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBackgroundHandler);
 return true;
}
Future<bool>  firebaseMessagingBackgroundHandler(RemoteMessage message) async {
 print("Running Background Message>>>");
 print("Recevied Background message ${message.messageId} -  ${message.toMap().toString()}");
 return true;
}
```

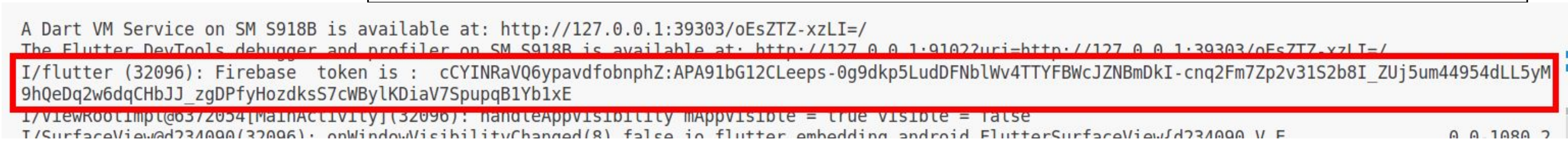
Firebase Services : Messaging

- **Firebase Mes**
  - **Mobile Ap**
    -



```
Future<bool> init_firebase_messaging() async {
 await firebase_requestPermission();
 FirebaseMessaging.onMessageOpenedApp.listen(_handleFirebaseMessage);

 await FirebaseMessaging.instance.getToken().then((token) {
   print("Firebase  token is :  $token");
   //Send it to Backend if you want to use it in the future ?
 });
 return true;
}

Future<bool> _handleFirebaseMessage(RemoteMessage message) async {
 print("\n\n\n#####Calling FireMessage Opening Handler..\n\n\n");
 return true;
}
```

03

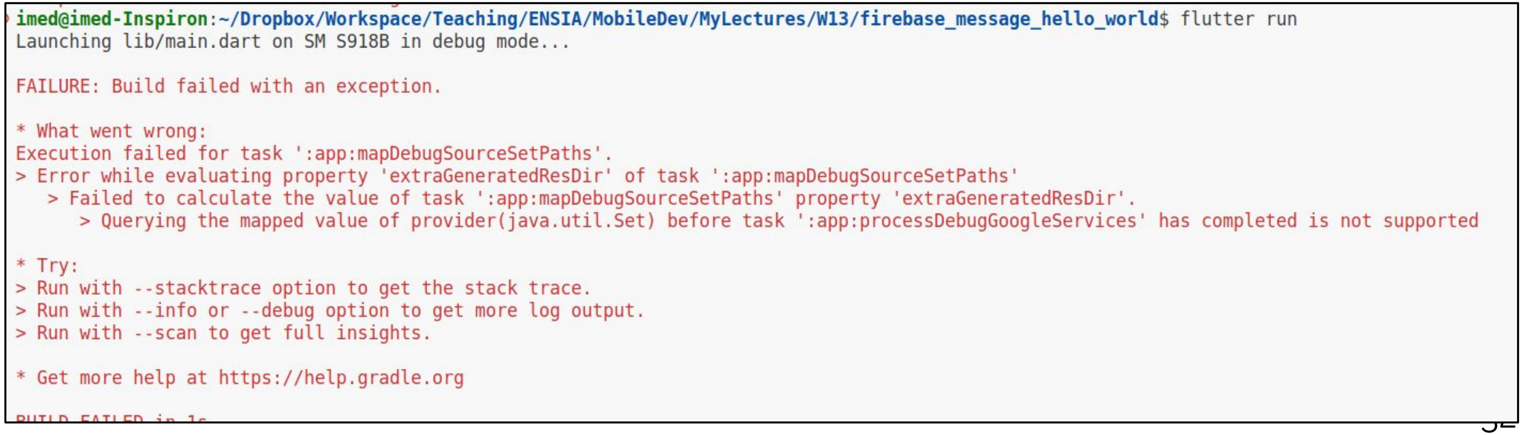


```
A Dart VM Service on SM S918B is available at: http://127.0.0.1:39303/oEsZTZ-xzLI=/
The Flutter DevTools debugger and profiler on SM S918B is available at: http://127.0.0.1:9102?uri=http://127.0.0.1:39303/oEsZTZ-xzLI=/
I/flutter (32096): Firebase  token is :  cCYINRaVQ6ypavdfobnphZ:APA91bG12CLeeps-0g9dkp5LudDFNblWv4TTYFBWcJZNBmDkI-cnq2Fm7Zp2v31S2b8I_ZUj5um44954dLL5yM
9hQeDq2w6dqCHbJJ_zgDPfyHozdksS7cWBylKDiaV7SpupqB1Yb1xE
I/ViewRootImpl@6372054[MainActivity](32096): handleAppVisibility mAppVisible = true visible = false
I/SurfaceView@d234090(32096): onWindowVisibilityChanged(8) false io.flutter.embedding.android.FlutterSurfaceView{d234090 V.E...... 0,0-1080,2
```

Firebase Services : Messaging

- **Firebase Messaging : Simple Steps for a Hello World !**
  - **Mobile App :**
    - Initialize the Firebase Instance : **Getting errors ?**

03



```
imed@imed-Inspiron:~/Dropbox/Workspace/Teaching/ENSIA/MobileDev/MyLectures/W13/firebase_message_hello_world$ flutter run
Launching lib/main.dart on SM S918B in debug mode...

FAILURE: Build failed with an exception.

* What went wrong:
Execution failed for task ':app:mapDebugSourceSetPaths'.
> Error while evaluating property 'extraGeneratedResDir' of task ':app:mapDebugSourceSetPaths'
   > Failed to calculate the value of task ':app:mapDebugSourceSetPaths' property 'extraGeneratedResDir'.
      > Querying the mapped value of provider(java.util.Set) before task ':app:processDebugGoogleServices' has completed is not supported

* Try:
> Run with --stacktrace option to get the stack trace.
> Run with --info or --debug option to get more log output.
> Run with --scan to get full insights.

* Get more help at https://help.gradle.org

BUILD FAILED in 1s
```

Firebase Services : Messaging

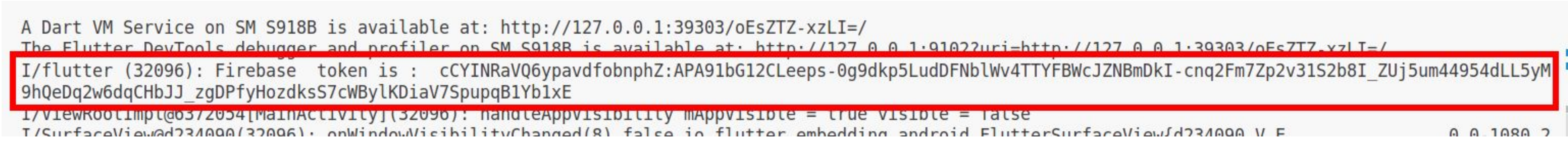
- **Firebase Messaging : Simple Steps for a Hello World !**
  - **Mobile App :**
    - Initialize the Firebase Instance : **Getting errors ?**
      - Update the Google Play Services
        - android/build.gradle
          - `classpath 'com.google.gms:google-services:4.4.0'`
      - Increase the minSdk
        - android/app/src/build.gradle
          - `minSdkVersion 19`

03

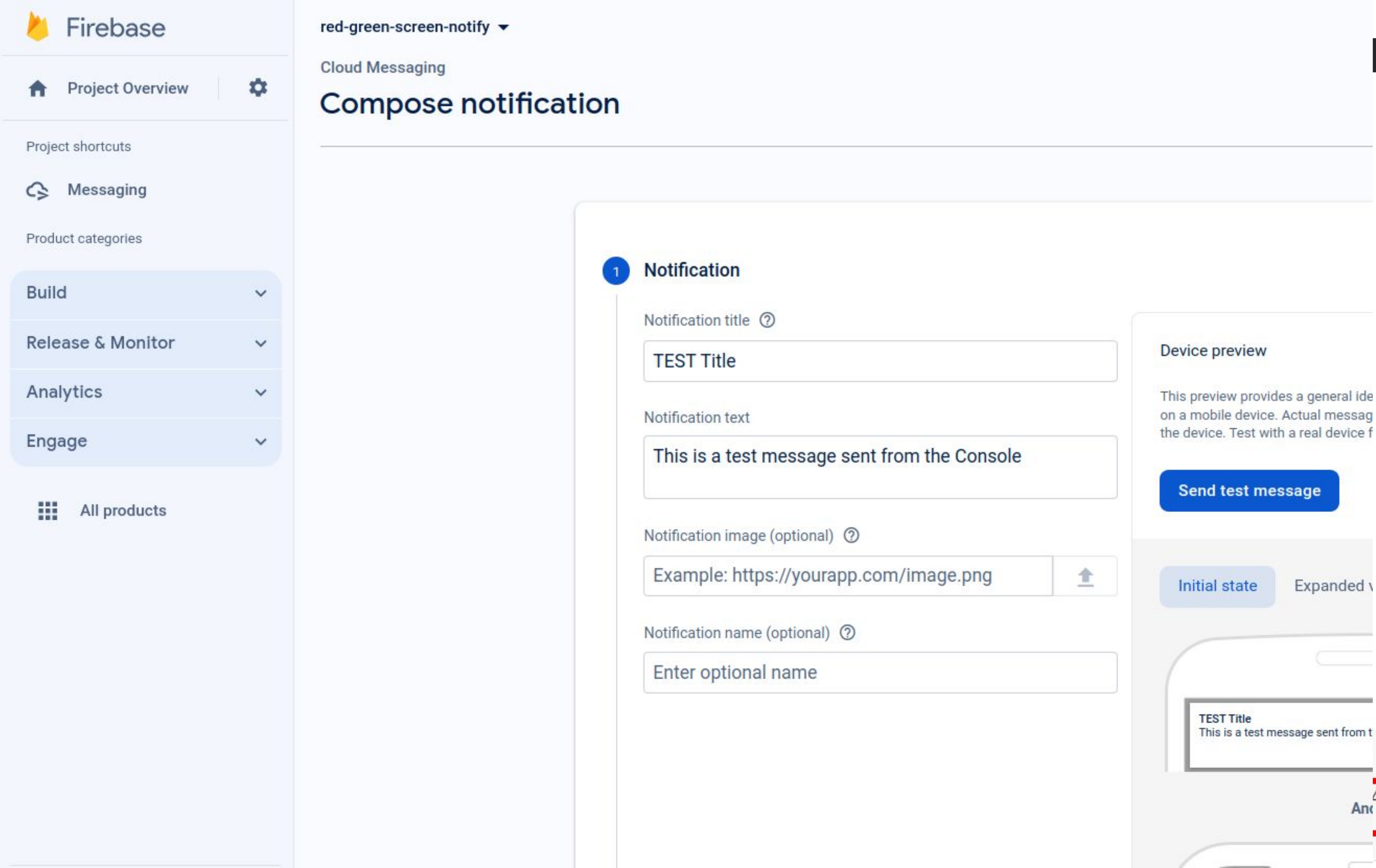
Firebase Services : Messaging

- **Firebase Messaging : Simple Steps for a Hello World !**
  - **Mobile App :**
    - Testing using Firebase Cloud Messaging Console :
      - Using the obtained token
      - We send a hello message

03



```
A Dart VM Service on SM S918B is available at: http://127.0.0.1:39303/oEsZTZ-xzLI=/
The Flutter DevTools debugger and profiler on SM S918B is available at: http://127.0.0.1:9102?uri=http://127.0.0.1:39303/oEsZTZ-xzLI=/
I/flutter (32096): Firebase  token is :  cCYINRaVQ6ypavdfobnphZ:APA91bG12CLeeps-0g9dkp5LudDFNblWv4TTYFBWcJZNBmDkI-cnq2Fm7Zp2v31S2b8I_ZUj5um44954dLL5yM
9hQeDq2w6dqCHbJJ_zgDPfyHozdksS7cWBylKDiaV7SpupqB1Yb1xE
I/ViewRootImpl@6372054[MainActivity](32096): handleAppVisibility mAppVisible = true visible = false
I/SurfaceView@d234090(32096): onWindowVisibilityChanged(8) false io.flutter.embedding.android.FlutterSurfaceView{d234090 V.E                    0,0-1080,2
```



Cloud Messaging

Compose notification

1 Notification

Notification title: TEST Title

Notification text: This is a test message sent from the Console

Notification image (optional): Example: https://yourapp.com/image.png

Notification name (optional): Enter optional name

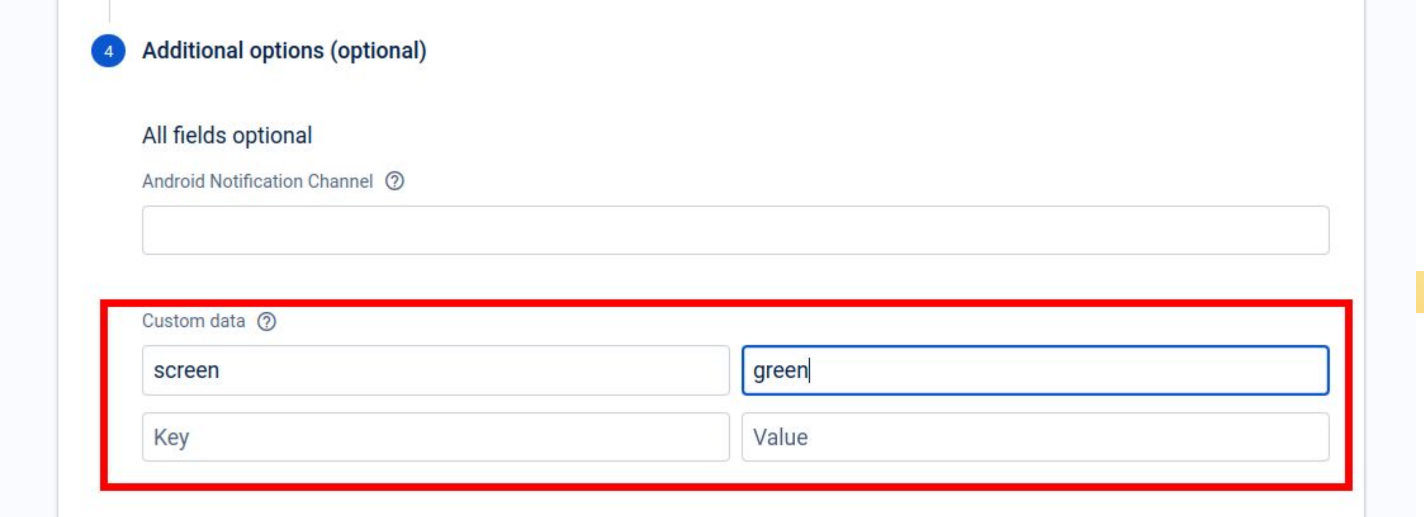
Device preview

This preview provides a general ide
on a mobile device. Actual messag
the device. Test with a real device f

```
A Dart
The Fl
I/flutt                                                            4dLL5yM
9hQeDq2
I/View
I/Surf                                                             1080 2
```

Fireba
Me

- **Firebase Mess**
  - **Mobile App**



4 Additional options (optional)

All fields optional

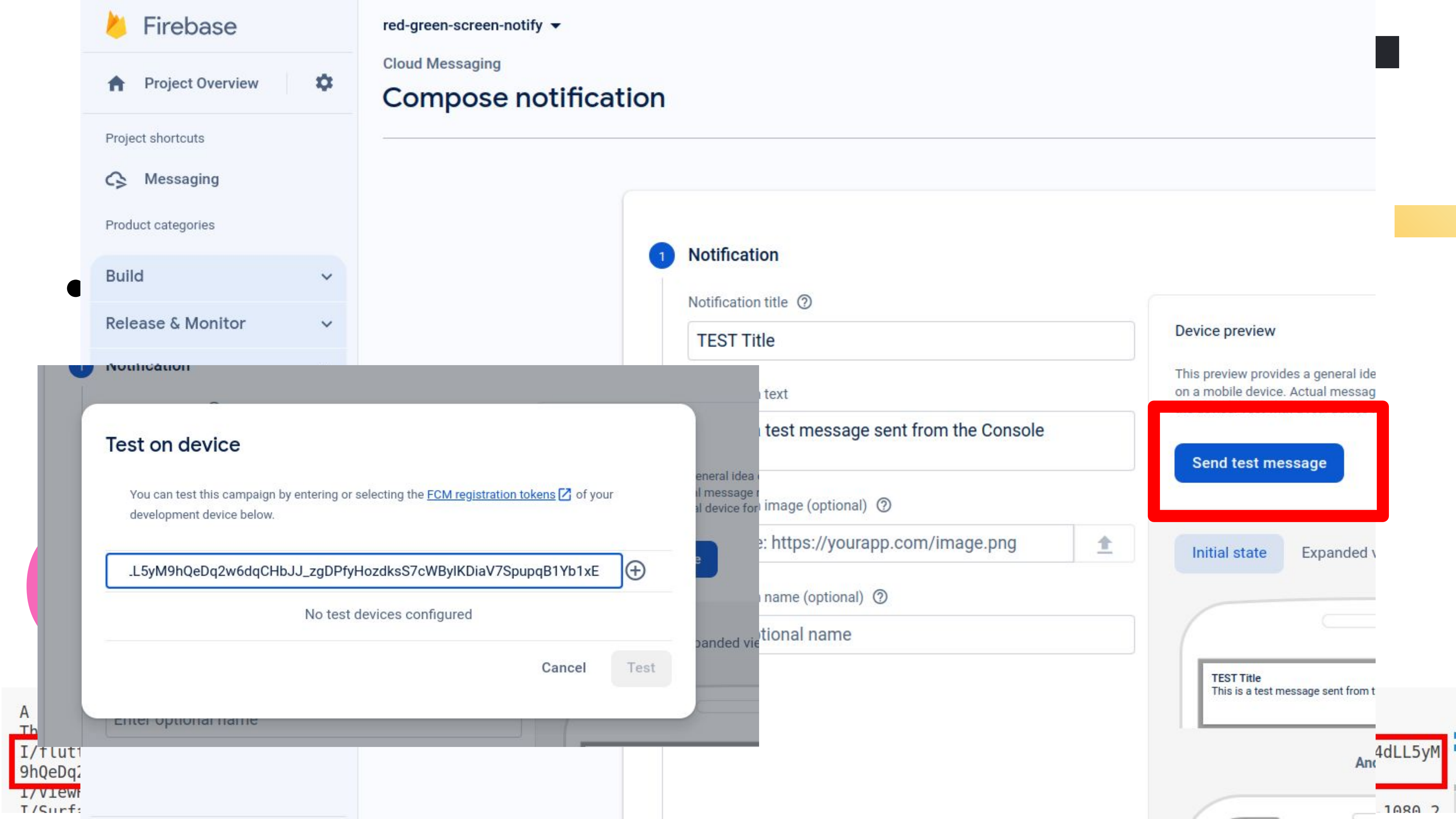
Android Notification Channel

Custom data

screen	green
Key	Value

03

- Testing using Firebase Cloud Messaging Console :
  - Beside the title and body, Firebase Message may (**ALWAYS**) have an extra data payload in the form a dictionary/map



red-green-screen-notify

Cloud Messaging

Compose notification

1 Notification

Notification title: TEST Title

…text: …test message sent from the Console

…image (optional): …: https://yourapp.com/image.png

…name (optional): …tional name

Device preview

This preview provides a general ide… on a mobile device. Actual messag…

Send test message

Initial state | Expanded v…

TEST Title
This is a test message sent from t…

Test on device

You can test this campaign by entering or selecting the FCM registration tokens of your development device below.

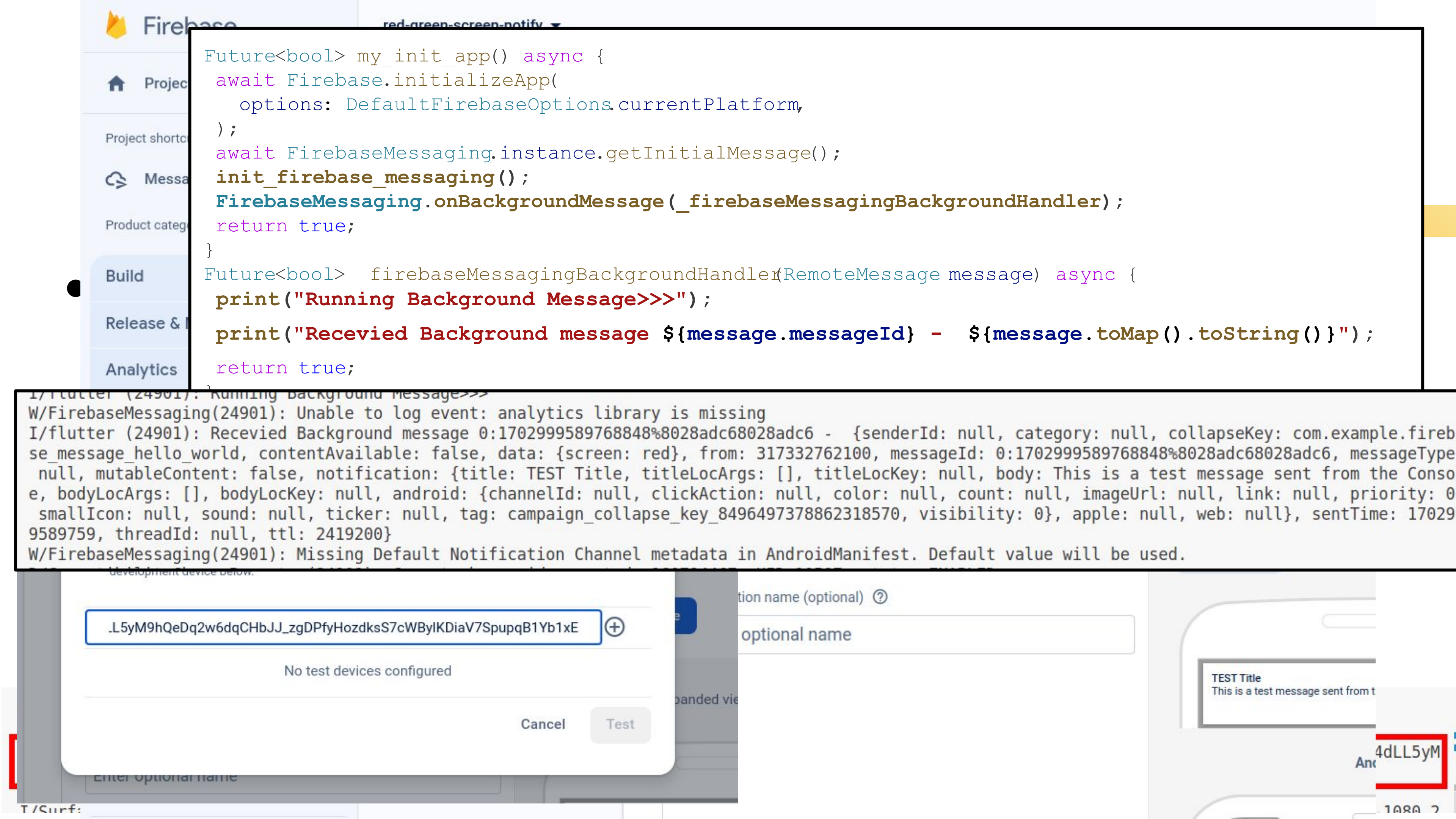
.L5yM9hQeDq2w6dqCHbJJ_zgDPfyHozdksS7cWBylKDiaV7SpupqB1Yb1xE

No test devices configured

Cancel | Test

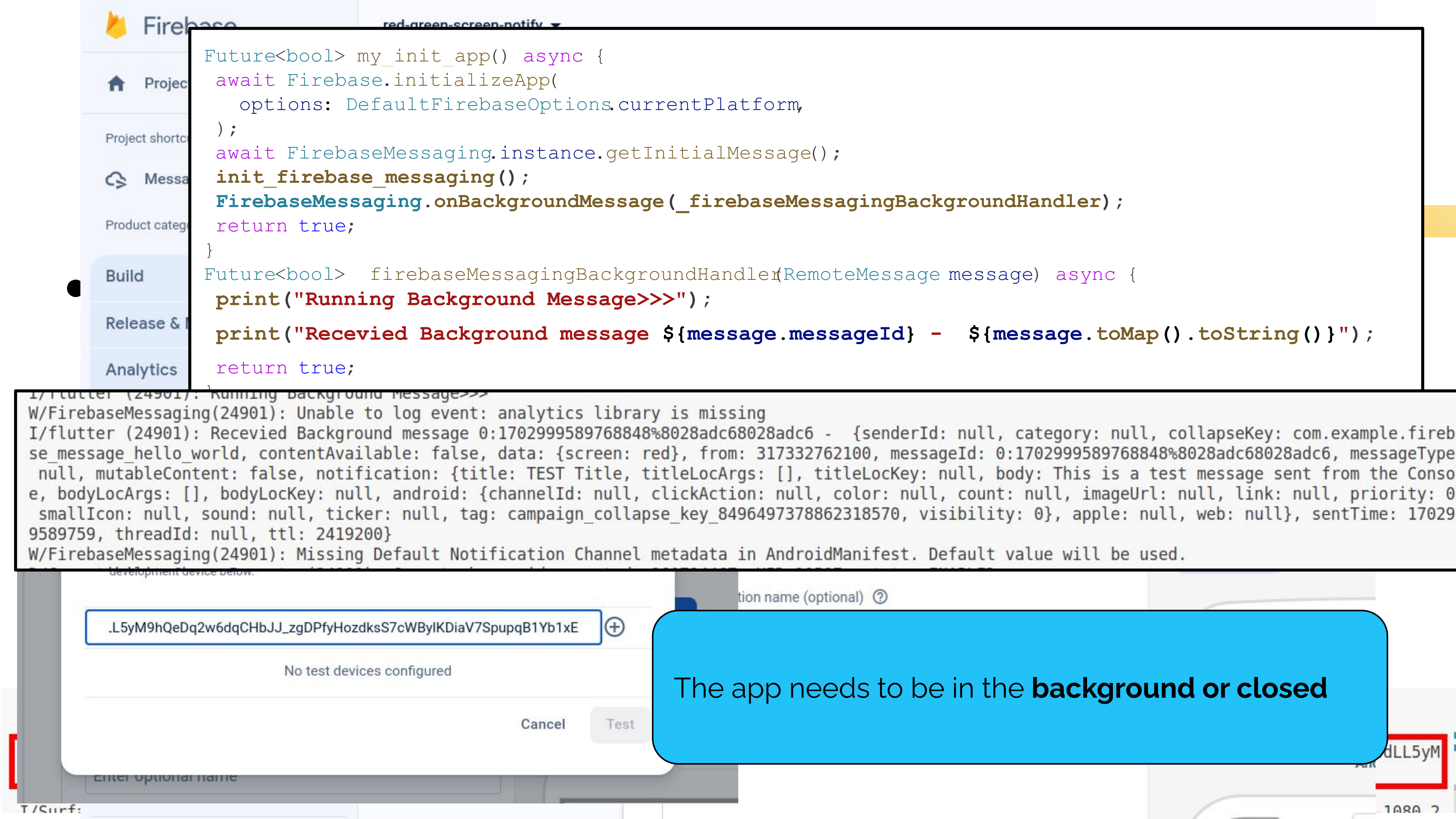
```
A
Th
I/flutt
9hQeDq2
I/Viewr
I/Surf
```

```
4dLL5yM
And
1080 2
```



```
Future<bool> my_init_app() async {
 await Firebase.initializeApp(
   options: DefaultFirebaseOptions.currentPlatform,
 );
 await FirebaseMessaging.instance.getInitialMessage();
 init_firebase_messaging();
 FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBackgroundHandler);
 return true;
}
Future<bool>  firebaseMessagingBackgroundHandler(RemoteMessage message) async {
 print("Running Background Message>>>");
 print("Recevied Background message ${message.messageId} -  ${message.toMap().toString()}");
 return true;
```

```
I/flutter (24901): Running Background Message>>>
W/FirebaseMessaging(24901): Unable to log event: analytics library is missing
I/flutter (24901): Recevied Background message 0:1702995897688488028adc68028adc6 -  {senderId: null, category: null, collapseKey: com.example.fireb
se_message_hello_world, contentAvailable: false, data: {screen: red}, from: 317332762100, messageId: 0:1702995897688488028adc68028adc6, messageType
 null, mutableContent: false, notification: {title: TEST Title, titleLocArgs: [], titleLocKey: null, body: This is a test message sent from the Conso
e, bodyLocArgs: [], bodyLocKey: null, android: {channelId: null, clickAction: null, color: null, count: null, imageUrl: null, link: null, priority: 0
 smallIcon: null, sound: null, ticker: null, tag: campaign_collapse_key_8496497378862318570, visibility: 0}, apple: null, web: null}, sentTime: 17029
9589759, threadId: null, ttl: 2419200}
W/FirebaseMessaging(24901): Missing Default Notification Channel metadata in AndroidManifest. Default value will be used.
```



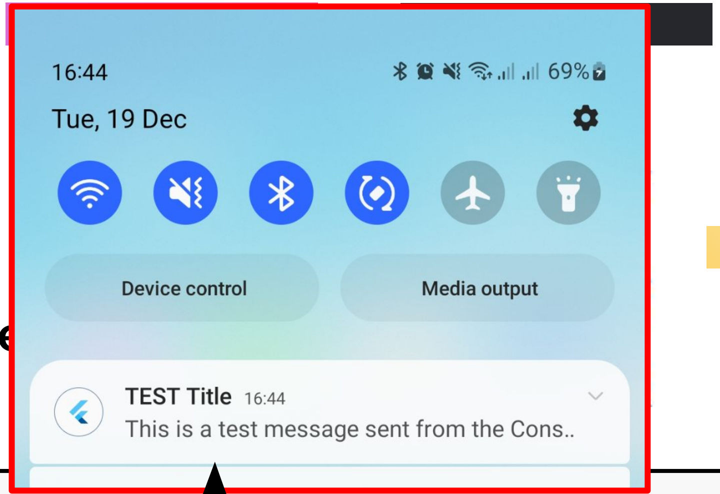
```
Future<bool> my_init_app() async {
 await Firebase.initializeApp(
   options: DefaultFirebaseOptions.currentPlatform,
 );
 await FirebaseMessaging.instance.getInitialMessage();
 init_firebase_messaging();
 FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBackgroundHandler);
 return true;
}
Future<bool>  firebaseMessagingBackgroundHandler(RemoteMessage message) async {
 print("Running Background Message>>>");
 print("Recevied Background message ${message.messageId} -  ${message.toMap().toString()}");
 return true;
```

```
I/flutter (24901): Running Background Message>>>
W/FirebaseMessaging(24901): Unable to log event: analytics library is missing
I/flutter (24901): Recevied Background message 0:1702999589768848%8028adc68028adc6 -  {senderId: null, category: null, collapseKey: com.example.fireb
se_message_hello_world, contentAvailable: false, data: {screen: red}, from: 317332762100, messageId: 0:1702999589768848%8028adc68028adc6, messageType
 null, mutableContent: false, notification: {title: TEST Title, titleLocArgs: [], titleLocKey: null, body: This is a test message sent from the Conso
e, bodyLocArgs: [], bodyLocKey: null, android: {channelId: null, clickAction: null, color: null, count: null, imageUrl: null, link: null, priority: 0
 smallIcon: null, sound: null, ticker: null, tag: campaign_collapse_key_8496497378862318570, visibility: 0}, apple: null, web: null}, sentTime: 17029
9589759, threadId: null, ttl: 2419200}
W/FirebaseMessaging(24901): Missing Default Notification Channel metadata in AndroidManifest. Default value will be used.
```

The app needs to be in the **background or closed**

Firebase Services : Messaging

- **Firebase Messaging : Simple Steps for a He**
  - **Mobile App :**



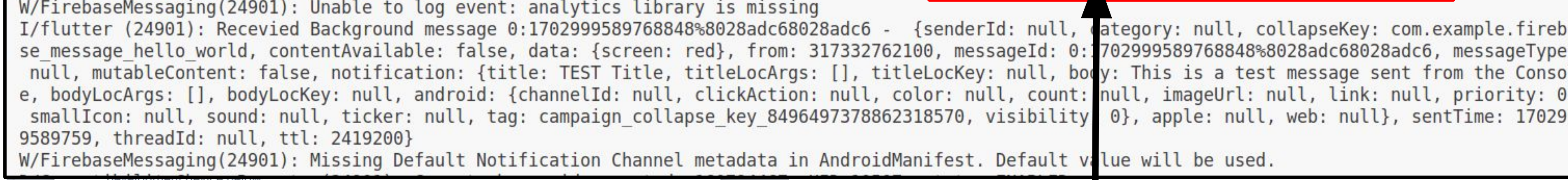
16:44

Tue, 19 Dec

69%

TEST Title 16:44

This is a test message sent from the Cons..



```
I/flutter (24901): Running Background Message>>>
W/FirebaseMessaging(24901): Unable to log event: analytics library is missing
I/flutter (24901): Recevied Background message 0:1702999589768848%8028adc68028adc6 -  {senderId: null, category: null, collapseKey: com.example.fireb
se_message_hello_world, contentAvailable: false, data: {screen: red}, from: 317332762100, messageId: 0:1702999589768848%8028adc68028adc6, messageType
 null, mutableContent: false, notification: {title: TEST Title, titleLocArgs: [], titleLocKey: null, body: This is a test message sent from the Conso
e, bodyLocArgs: [], bodyLocKey: null, android: {channelId: null, clickAction: null, color: null, count: null, imageUrl: null, link: null, priority: 0
 smallIcon: null, sound: null, ticker: null, tag: campaign_collapse_key_8496497378862318570, visibility: 0}, apple: null, web: null}, sentTime: 17029
9589759, threadId: null, ttl: 2419200}
W/FirebaseMessaging(24901): Missing Default Notification Channel metadata in AndroidManifest. Default value will be used.
```

development device below.

.L5yM9hQeDq2w6dqCHbJJ_zgDPfyHozdksS7cWBylKDiaV7SpupqB1Yb1xE

No test devices configured

```
dLL5yM
= true visible = false
I/SurfaceView@d234090(32096): onWindowVisibilityChanged(8) false io.flutter.embedding.android.FlutterSurfaceView{d234090 V.E
```

What happens when they click on the notification message ?

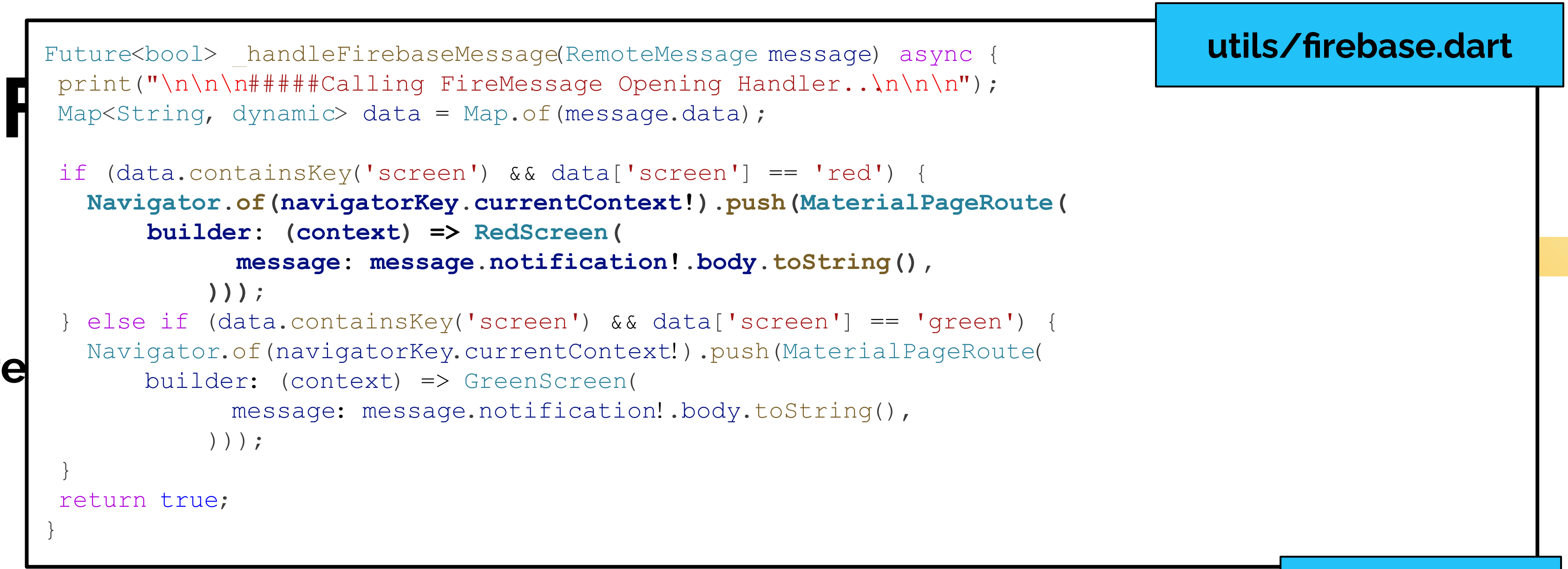
Firebase Services : Messaging

- **Firebase Messaging : Simple Steps for a Hello World !**
    - **Mobile App :**
        - Custom Navigation from Background Notification:
            - If inside the data payload,
                - the screen value is red ⇒ show red screen
                - the screen value is green ⇒ show green screen
                - Otherwise, just do the default behaviour

04

F

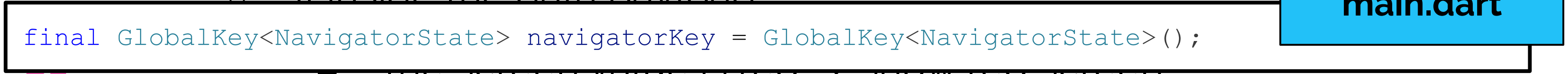
utils/firebase.dart



```
Future<bool> _handleFirebaseMessage(RemoteMessage message) async {
 print("\n\n\n#####Calling FireMessage Opening Handler..\n\n\n");
 Map<String, dynamic> data = Map.of(message.data);

 if (data.containsKey('screen') && data['screen'] == 'red') {
   Navigator.of(navigatorKey.currentContext!).push(MaterialPageRoute(
       builder: (context) => RedScreen(
             message: message.notification!.body.toString(),
           )));
 } else if (data.containsKey('screen') && data['screen'] == 'green') {
   Navigator.of(navigatorKey.currentContext!).push(MaterialPageRoute(
       builder: (context) => GreenScreen(
             message: message.notification!.body.toString(),
           )));
 }
 return true;
}
```

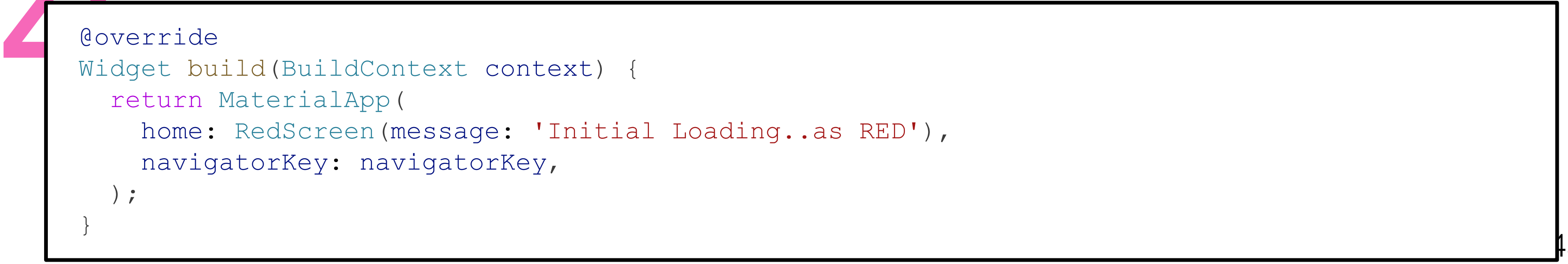
- Fire
  - ○
  - ○ If inside the data payload
    - the screen value is red → show red screen

main.dart



```
final GlobalKey<NavigatorState> navigatorKey = GlobalKey<NavigatorState>();
```

04



```
@override
Widget build(BuildContext context) {
  return MaterialApp(
    home: RedScreen(message: 'Initial Loading..as RED'),
    navigatorKey: navigatorKey,
  );
}
```

Firebase Services : Messaging

- **Firebase Messaging : Simple Steps for a Hello World !**
  - **Mobile App :**
    - Receiving In-App Messages :
      - Initialize the Local Notification Plugin Instance
      - Need to listen onMessage from Firebase
      - Integrate Local Notification Plugin to show message
      - Handle Navigation if needed.

05

Firebase Services : Messaging

main.dart

```
FlutterLocalNotificationsPlugin flutterLocalNotificationsPlugin =
   FlutterLocalNotificationsPlugin();
```

- **Firebase Messaging : Simple Steps for a Hello World !**

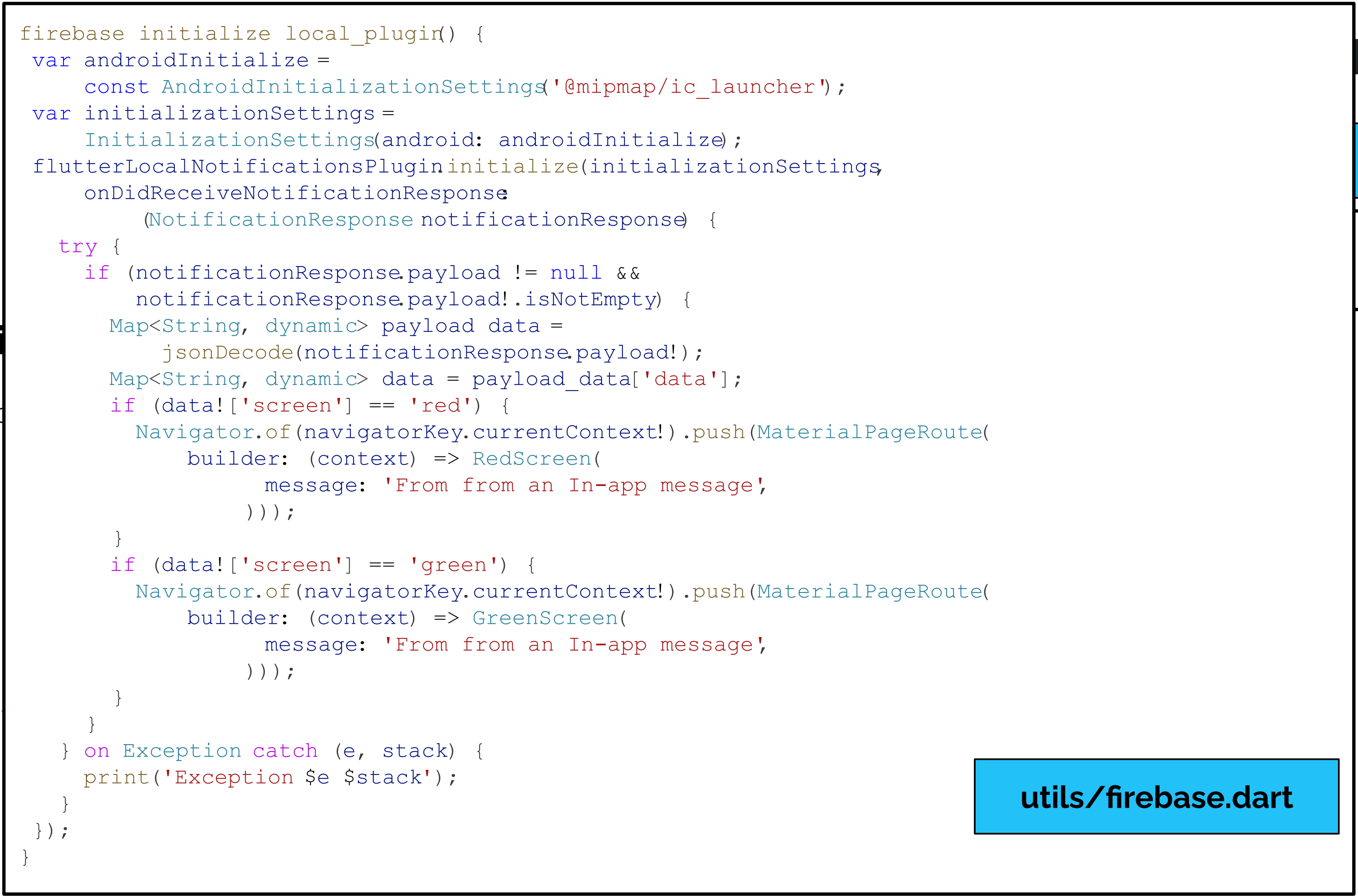
utils/firebase.dart

```
Future<bool> init firebase messaging() async {
 await firebase_requestPermission();
 await firebase_initialize_local_plugin();
 . . .
}
```

05

  - Integrate Local Notification Plugin to show message
  - Handle Navigation if needed.

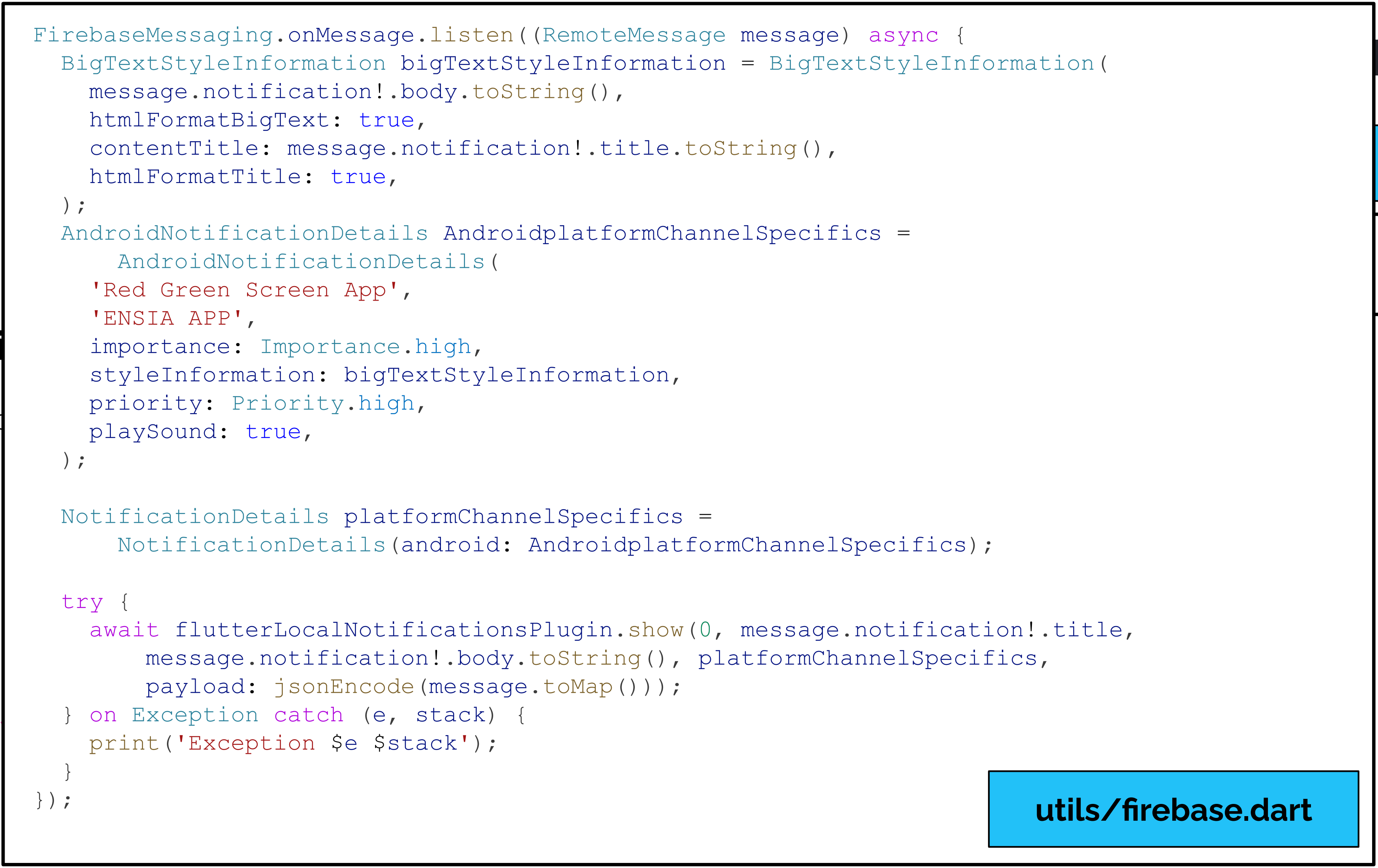
- Fi



```
firebase initialize local_plugin() {
 var androidInitialize =
     const AndroidInitializationSettings('@mipmap/ic_launcher');
 var initializationSettings =
     InitializationSettings(android: androidInitialize);
 flutterLocalNotificationsPlugin.initialize(initializationSettings,
     onDidReceiveNotificationResponse:
         (NotificationResponse notificationResponse) {
   try {
     if (notificationResponse.payload != null &&
         notificationResponse.payload!.isNotEmpty) {
       Map<String, dynamic> payload data =
           jsonDecode(notificationResponse.payload!);
       Map<String, dynamic> data = payload_data['data'];
       if (data!['screen'] == 'red') {
         Navigator.of(navigatorKey.currentContext!).push(MaterialPageRoute(
             builder: (context) => RedScreen(
                   message: 'From from an In-app message',
                 )));
       }
       if (data!['screen'] == 'green') {
         Navigator.of(navigatorKey.currentContext!).push(MaterialPageRoute(
             builder: (context) => GreenScreen(
                   message: 'From from an In-app message',
                 )));
       }
     }
   } on Exception catch (e, stack) {
     print('Exception $e $stack');
   }
 });
}
```

utils/firebase.dart

- Fi



```
FirebaseMessaging.onMessage.listen((RemoteMessage message) async {
  BigTextStyleInformation bigTextStyleInformation = BigTextStyleInformation(
    message.notification!.body.toString(),
    htmlFormatBigText: true,
    contentTitle: message.notification!.title.toString(),
    htmlFormatTitle: true,
  );
  AndroidNotificationDetails AndroidplatformChannelSpecifics =
      AndroidNotificationDetails(
    'Red Green Screen App',
    'ENSIA APP',
    importance: Importance.high,
    styleInformation: bigTextStyleInformation,
    priority: Priority.high,
    playSound: true,
  );

  NotificationDetails platformChannelSpecifics =
      NotificationDetails(android: AndroidplatformChannelSpecifics);

  try {
    await flutterLocalNotificationsPlugin.show(0, message.notification!.title,
        message.notification!.body.toString(), platformChannelSpecifics,
        payload: jsonEncode(message.toMap()));
  } on Exception catch (e, stack) {
    print('Exception $e $stack');
  }
});
```

utils/firebase.dart

Firebase Services : Messaging

06

- **Firebase Messaging : Simple Steps for a Hello World !**
  - **Mobile App :**
    - Subscribing users to **Topics** :
      - When the user visits The red screen, we subscribe them to the **Red** Topic and the same for **Green.**
      - Instructions of Firebase Code :
        - **await FirebaseMessaging.instance.subscribeToTopic("Red");**
        - **await FirebaseMessaging.instance.unsubscribeFromTopic("Green”);**

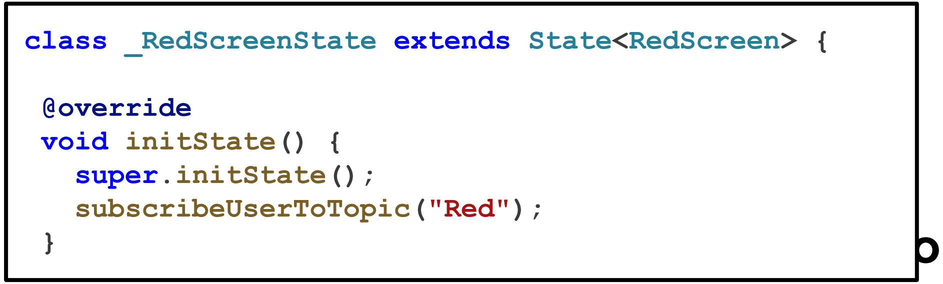
Firebase Services : Messaging

Revision : Where to inject those two lines of code ?

- **Firebase Messaging : Simple Steps for a Hello World !**
  - **Mobile App :**
    - Subscribing users to **Topics** :
      - When the user visits The red screen, we subscribe them to the **Red** Topic and the same for **Green.**
      - Instructions of Firebase Code :
        - **await FirebaseMessaging.instance.subscribeToTopic("Red");**
        - **await FirebaseMessaging.instance.unsubscribeFromTopic("Green”);**

06

Firebase Services :

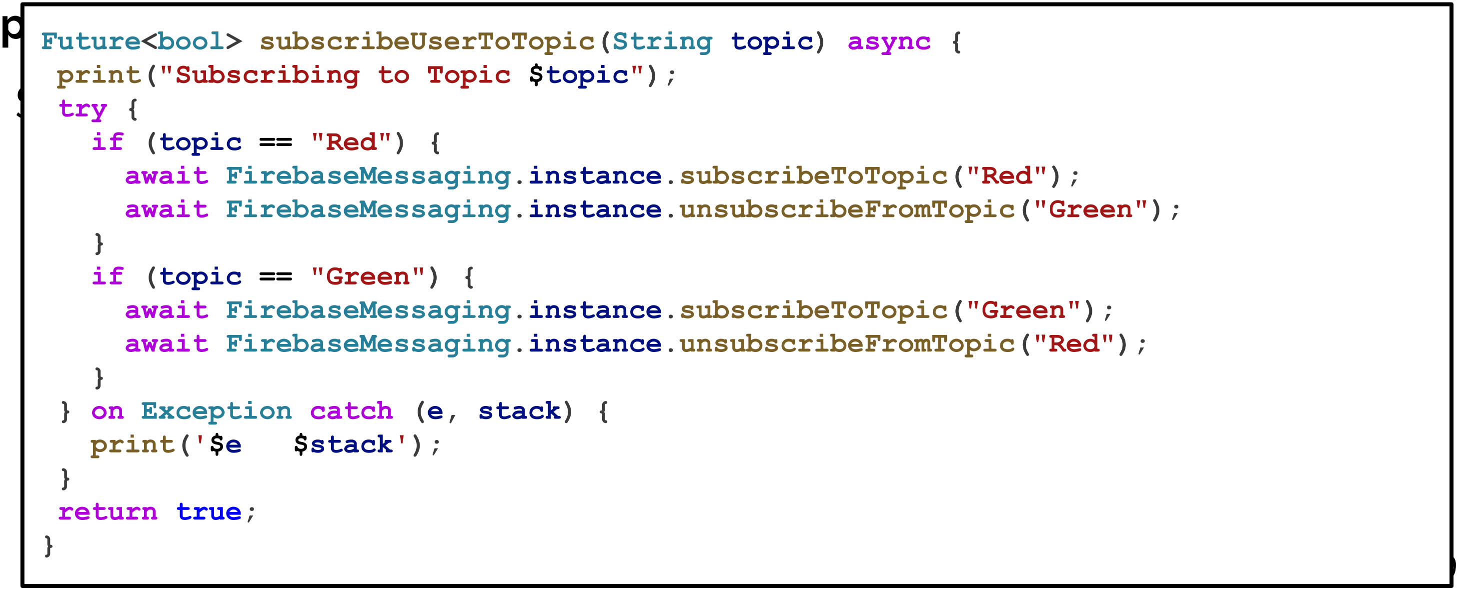


```
class _RedScreenState extends State<RedScreen> {

 @override
 void initState() {
   super.initState();
   subscribeUserToTopic("Red");
 }
```

or a Hello World !

○ Mobile Ap

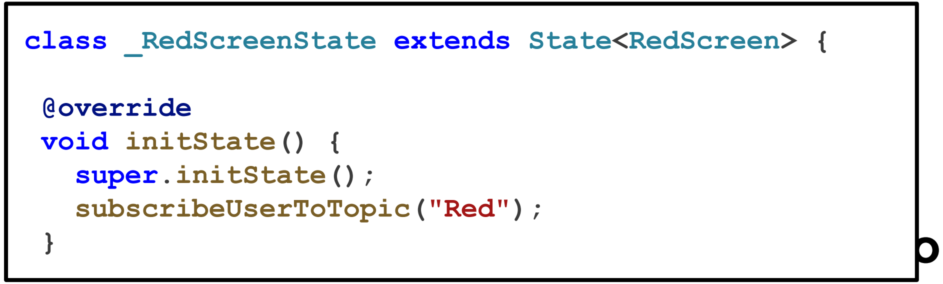
● S



```
Future<bool> subscribeUserToTopic(String topic) async {
 print("Subscribing to Topic $topic");
 try {
   if (topic == "Red") {
     await FirebaseMessaging.instance.subscribeToTopic("Red");
     await FirebaseMessaging.instance.unsubscribeFromTopic("Green");
   }
   if (topic == "Green") {
     await FirebaseMessaging.instance.subscribeToTopic("Green");
     await FirebaseMessaging.instance.unsubscribeFromTopic("Red");
   }
 } on Exception catch (e, stack) {
   print('$e   $stack');
 }
 return true;
}
```

06

Firebase Services :

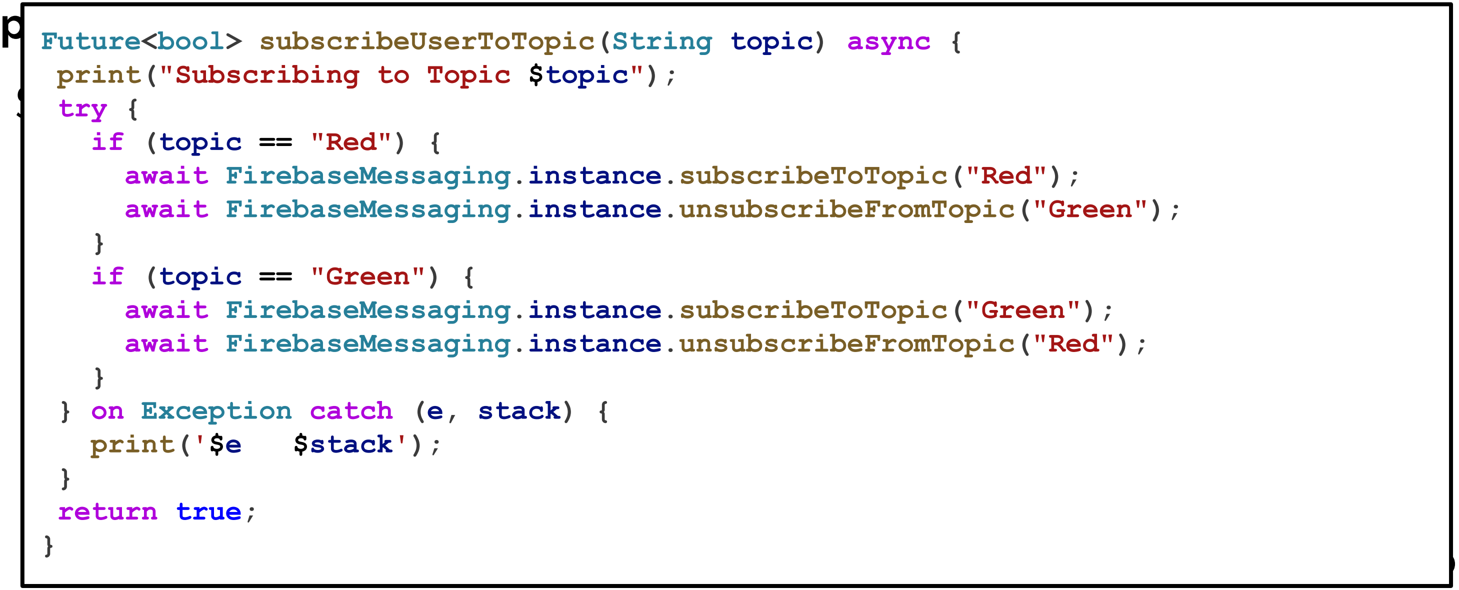


```
class _RedScreenState extends State<RedScreen> {

 @override
 void initState() {
   super.initState();
   subscribeUserToTopic("Red");
 }
```

How to test sending to specific topic ?

or a Hello World !

○ Mobile Ap



```
Future<bool> subscribeUserToTopic(String topic) async {
 print("Subscribing to Topic $topic");
 try {
   if (topic == "Red") {
     await FirebaseMessaging.instance.subscribeToTopic("Red");
     await FirebaseMessaging.instance.unsubscribeFromTopic("Green");
   }
   if (topic == "Green") {
     await FirebaseMessaging.instance.subscribeToTopic("Green");
     await FirebaseMessaging.instance.unsubscribeFromTopic("Red");
   }
 } on Exception catch (e, stack) {
   print('$e   $stack');
 }
 return true;
}
```

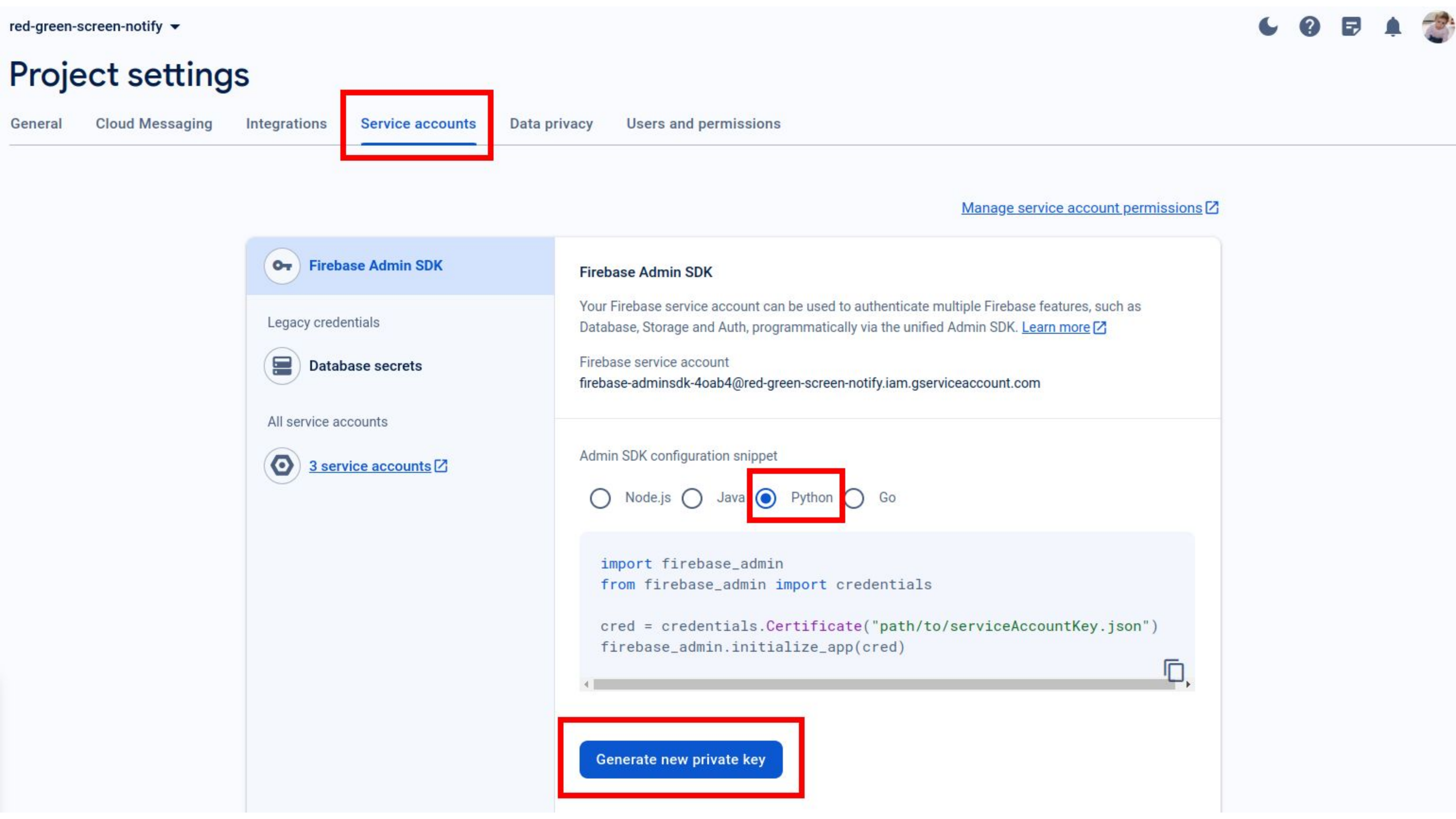
06

Firebase Services : Messaging

- **Firebase Messaging : Simple Steps for a Hello World !**
  - **Testing the Message :**
    - If you are using Firebase Console, you need to be patient as it takes time.
    - Prefer to use Python Client

Firebase Services : Messaging

- **Firebase Messaging : Simple Steps for a Hello World !**
    - **Testing the Message :**
        - Steps for testing using a Hello World Python Client
            - Install the FirebaseSDK for your preferred language (Java, Python ..)
                - Python : **sudo pip3 install firebase_admin**
            - Download the Credentials JSON file from your project
            - Write the Hello World



Project settings

Firebase Admin SDK

Your Firebase service account can be used to authenticate multiple Firebase features, such as Database, Storage and Auth, programmatically via the unified Admin SDK. Learn more

Firebase service account
firebase-adminsdk-4oab4@red-green-screen-notify.iam.gserviceaccount.com

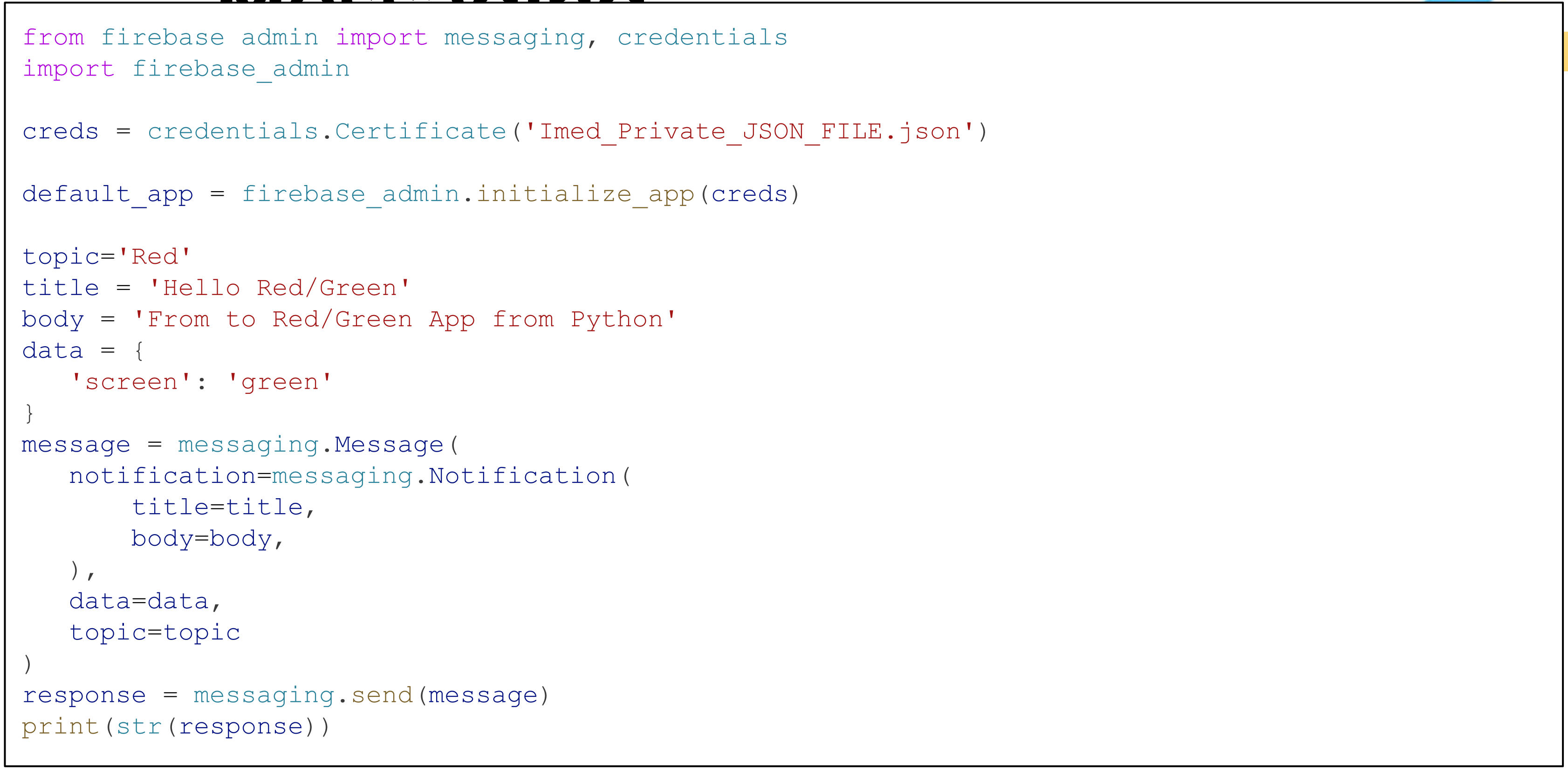
Admin SDK configuration snippet

○ Node.js ○ Java ● Python ○ Go

```
import firebase_admin
from firebase_admin import credentials

cred = credentials.Certificate("path/to/serviceAccountKey.json")
firebase_admin.initialize_app(cred)
```

Firebase Services : Messaging

-



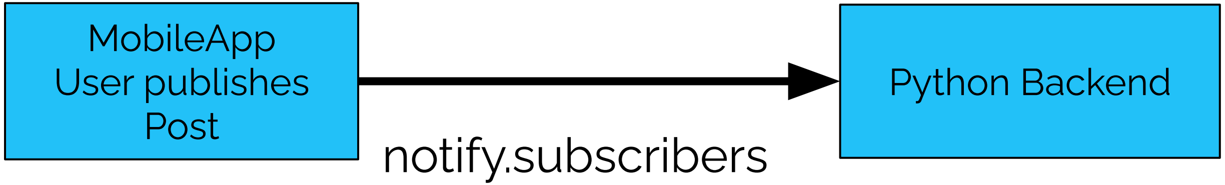
```
from firebase admin import messaging, credentials
import firebase_admin

creds = credentials.Certificate('Imed_Private_JSON_FILE.json')

default_app = firebase_admin.initialize_app(creds)

topic='Red'
title = 'Hello Red/Green'
body = 'From to Red/Green App from Python'
data = {
   'screen': 'green'
}
message = messaging.Message(
   notification=messaging.Notification(
       title=title,
       body=body,
   ),
   data=data,
   topic=topic
)
response = messaging.send(message)
print(str(response))
```

Firebase Services : Messaging

- **Firebase Messaging : How to Integrate with my Project**
  - **Case Study :**
    - **ENSIA Social App :**
      - When someone publishes a post to a specific category,
        - All users following that app, will receive a notification about this post
        - When, they click on the notification box, it will take them to another screen.

Firebase Services : Messaging

- **Firebase Messaging : How to Integrate with my Project**
  - **Case Study :**
    - **ENSIA Social App :**



MobileApp
User publishes
Post

notify.subscribers

Python Backend

If you are Certain that the execution will be quick : Send to Firebase Servers directly.

Otherwise, queue approach, store into a database/queue and process them incrementally

Firebase Services : Messaging

- **Firebase Messaging : How to Integrate with my Project**
    - **Case Study :**
        - **All Projects :**
            - You release a new version of the App, you want to send kind message for all users to download to upgrade to the latest version ?
            - You have some news to share with your users ?

Firebase Services : Messaging

- **Firebase Messaging : How to Integrate with my Project**
  - **Case Study :**
    - **All Projects :**
      - You release a new version of the App, you want to send kind message for all users to download to upgrade to the latest version ?
      - You have some news to share with your users ?
      - Users **who didn't use the app for a month or more,** send them a promotional message to get them back to use your app

Firebase Services : Messaging

- **Firebase Messaging : How to learn ?**
  - Here is the MVP without Messaging Integration
    - Link to Dropbox here ?
  - **Add the Firebase Cloud Messaging to it on your own**
  - Final App with messaging Integrated:
    - Link to Dropbox here..

Section 3

Firestore

Firebase Services :
Cloud Firestore Database

- **Cloud Firestore Database :**
  - Flexible, scalable database for mobile, web, and server development.
  - It is Like Firebase Realtime Database, it keeps your data in sync across client apps through realtime listeners.
  - It offers offline support for mobile and web so you can build responsive apps that work regardless of network latency or Internet connectivity.

Firebase Services :
Cloud Firestore Database

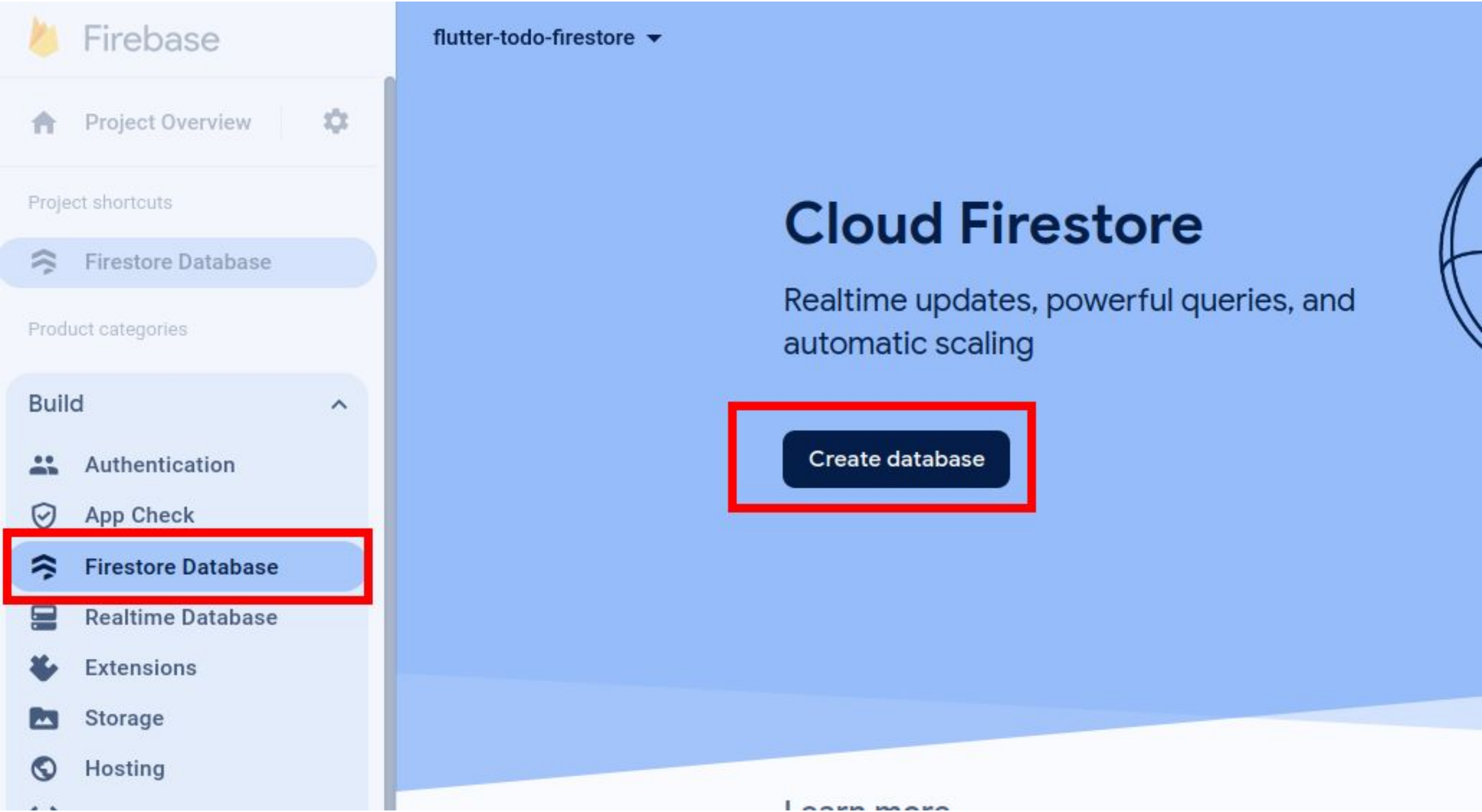
- **Cloud Firestore Database :**
    - Storage is structured as :
        - List of Collections
            - Each Collection contains Document
                - Document is a simple JSON text/Map.

Firebase Services :
Cloud Firestore Database

- **Cloud Firestore Database :**
  - Getting Started : Case Study : To-do app (No need for user authentication)
    - Decide whether Firestore is relevant to your mobile app.
    - Can integrate with other software components : Backend ?
    - Define the schema/structuring of the data.
    - Define the security policy to get to your data.
    - Integrate with your App.

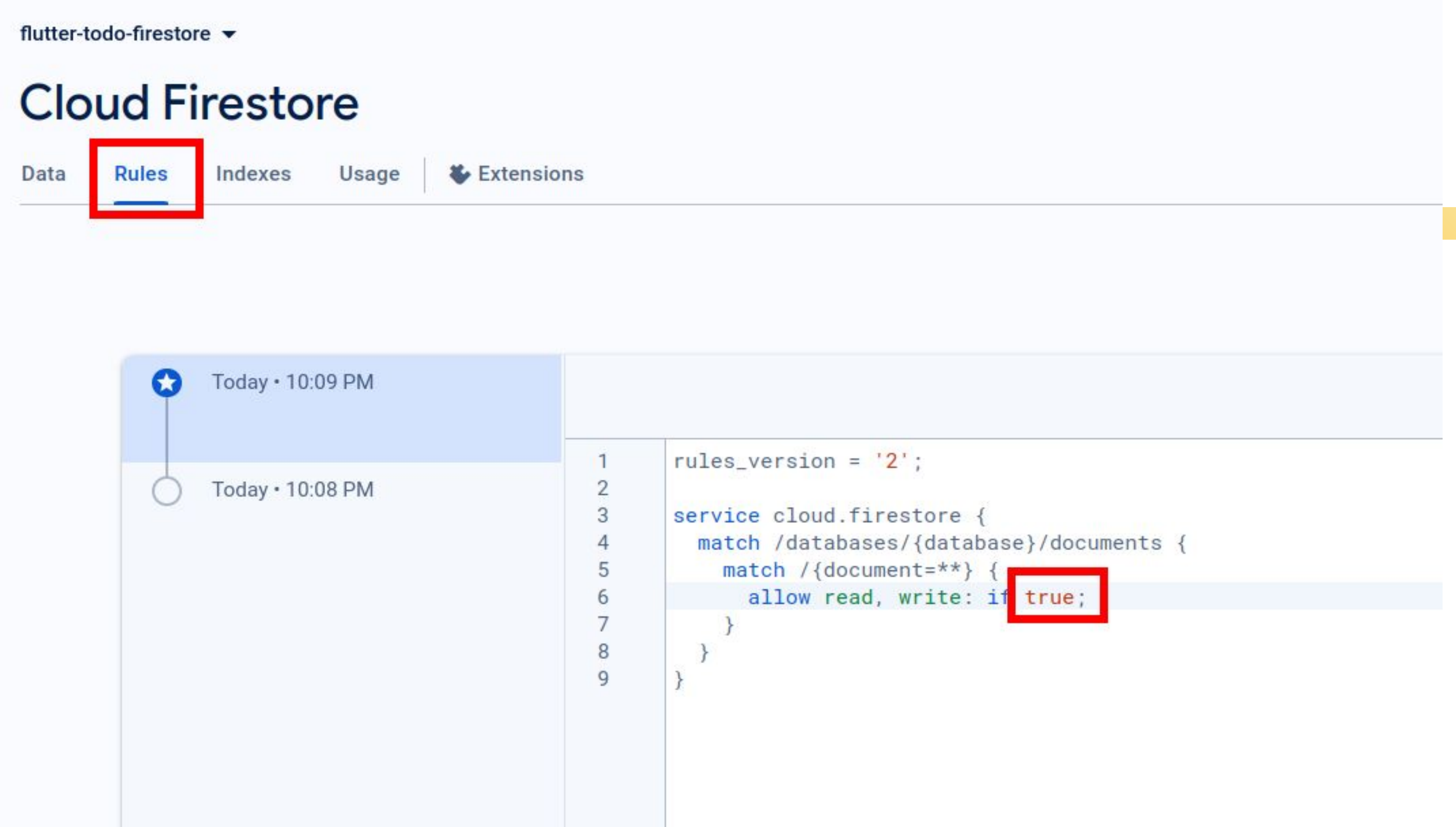
Firebase Services :
Cloud Firestore Database

- **Cloud Firestore Database :**
  - Integration with the App :
    - **flutterfire configure**
    - **flutter pub add firebase_core**
    - **flutter pub add cloud_firestore**



Cloud Firestore

Realtime updates, powerful queries, and automatic scaling

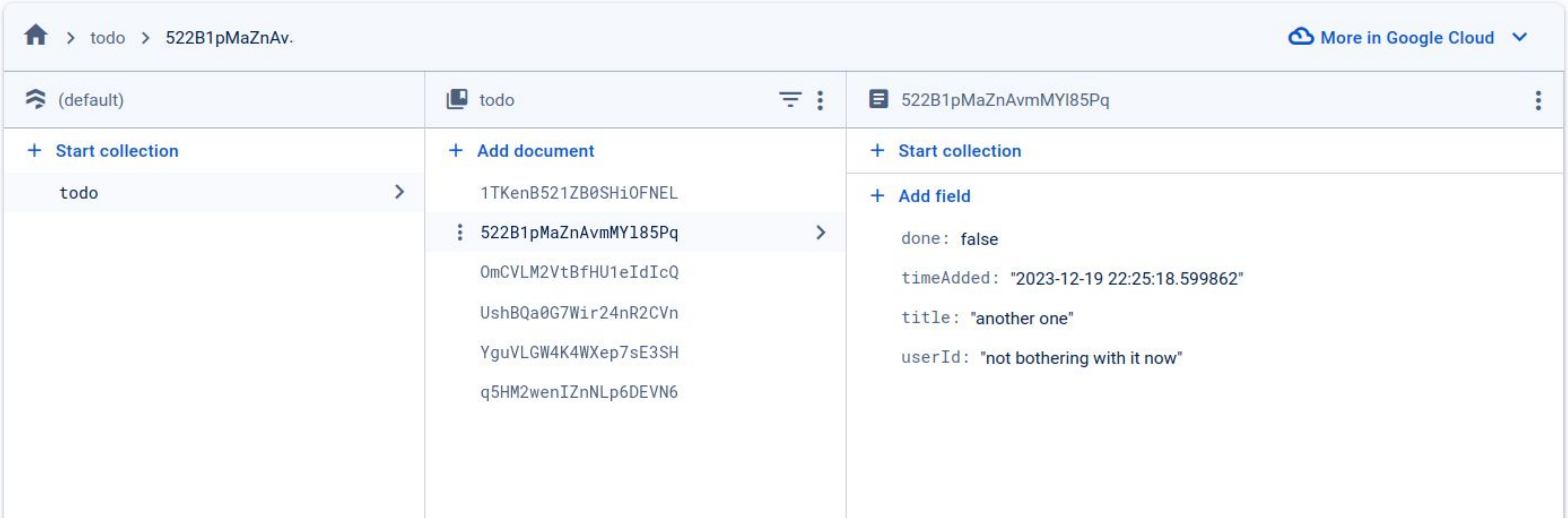


Cloud Firestore

```
rules_version = '2';

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

Firebase Services :
Cloud Firestore Database

- Cloud Firestore Database :



todo

1TKenB521ZB0SHiOFNEL
522B1pMaZnAvmMY185Pq
OmCVLM2VtBfHU1eIdIcQ
UshBQa0G7Wir24nR2CVn
YguVLGW4K4WXep7sE3SH
q5HM2wenIZnNLp6DEVN6

```
done: false
timeAdded: "2023-12-19 22:25:18.599862"
title: "another one"
userId: "not bothering with it now"
```

Section 4

Firebase Remote Config

Firebase Services : Remote Config

- **Features :**
  - Set Variables with Values from your Firebase Dashboard Console.
  - Mobile Apps will fetch such values and the business logic can be performed accordingly.
  - Variables values can be set and personalized differently based on different parameters related to the user (Region, Language...)

Firebase Services : Remote Config

- **Remote Config Hello World !**
  - Will be shared on Dropbox.

Lecture Demo Apps

- Firebase Database Storage
    - https://www.dropbox.com/scl/fo/060ia76ms06tlcxwhmkaw/h?rlkey=w0083imjbn84c4g5w5e917sh3&dl=0
- Firebase Messaging
    - Mobile App
    - Backend Message
- Firestore To do application:
    -

Resources

- https://firebase.google.com/docs/auth/admin/verify-id-tokens#retrieve_id_tokens_on_clients
- https://firebase.google.com/docs/firestore/quickstart#dart