

NOTIFICATIONS

in Flutter (Android)



@itsvinayyay

```
class NotificationServices {
  FirebaseMessaging messaging =
  FirebaseMessaging.instance;
  final FlutterLocalNotificationsPlugin
  _flutterLocalNotificationsPlugin =
    FlutterLocalNotificationsPlugin();

  void requestNotificationPermission() async {
    NotificationSettings settings = await
messaging.requestPermission(
      alert: true,
      announcement: true,
      badge: true,
      carPlay: true,
      criticalAlert: true,
      provisional: true,
      sound: true,
    );

    if (settings.authorizationStatus ==
AuthorizationStatus.authorized) {
      print("User Granted Permission");
    } else if (settings.authorizationStatus ==
AuthorizationStatus.provisional) {
      print("User Granted Provisional Permission");
    } else {
      print("User Denied Permission");
    }
  }

  void initLocalNotifications(BuildContext context,
RemoteMessage message) async {
    var androidInitializationSettings =
AndroidInitializationSettings('@mipmap/ic_launcher');
```

```

    var initializationSetting = InitializationSettings(
      android: androidInitializationSettings,
    );

    await
_flutterLocalNotificationsPlugin.initialize(initializationSetting,
      onDidReceiveNotificationResponse: (payload) {
        handleMessage(context, message);
      });
  }

void firebaseInit(BuildContext context) {
  FirebaseMessaging.onMessage.listen((message) {
    print(message.notification!.title.toString());
    print(message.notification!.body.toString());
    print(message.data.toString());
    print(message.data['message'].toString());
    print(message.data['id'].toString());

    if(Platform.isAndroid){
      initLocalNotifications(context, message);
    }
    showNotification(message);
  });
}

Future<void> showNotification(RemoteMessage message)
async {
  AndroidNotificationChannel channel =
  AndroidNotificationChannel(
    Random.secure().nextInt(100000).toString(),
    "High Importance Notification",
    importance: Importance.max);

```

```

        AndroidNotificationDetails androidNotificationDetails
=
        AndroidNotificationDetails(channel.id.toString(),
channel.name.toString(),
        channelDescription: "Your channel
Description",
        importance: Importance.high,
        priority: Priority.high,
        ticker: 'ticker',
        icon: '@mipmap/ic_launcher',
    );

    NotificationDetails notificationDetails =
        NotificationDetails(android:
androidNotificationDetails);

    Future.delayed(Duration.zero, () {
        _flutterLocalNotificationsPlugin.show(
            0,
            message.notification!.title.toString(),
            message.notification!.body.toString(),
            notificationDetails);
    });
}

Future<String> getDeviceToken() async {
    String? token = await messaging.getToken();
    return token!;
}

void isTokenRefresh() async {
    messaging.onTokenRefresh.listen((event) {
        event.toString();
    });
}

```

```
Future<void> setupInteractMessage(BuildContext
context)async{

    //when app is terminated
    RemoteMessage? initialMessage = await
FirebaseMessaging.instance.getInitialMessage();

    if(initialMessage!=null){
        handleMessage(context, initialMessage);
    }

    //when app is in background
    FirebaseMessaging.onMessageOpenedApp.listen((event) {
        handleMessage(context, event);
    });
}

void handleMessage(BuildContext context, RemoteMessage
message){
    if(message.data['message'] == 'Hello'){
        Navigator.pushNamed(context,
Routes.notificationsScreen);
    }
}
}
```

NOTIFICATION SERVICES

Requests permission from user

REQUEST NOTIFICATIONS PERMISSION

Specifies different types of notifications like alerts, badges, and sounds.

SHOW NOTIFICATIONS

Uses FlutterLocalNotificationsPlugin to display the notification with a title and body.

SETUP INTERACT MESSAGE

used to handle messages when the app is terminated or in the background.

IS TOKEN REFRESH

Listens for token refresh events. If the device's token changes, this method will be called.

GET DEVICE TOKEN

Retrieves the device's unique token, which can be used to send notifications to this specific device.

FIREBASE INIT

Initializes Firebase messaging and listens for incoming messages.

INIT LOCAL NOTIFICATIONS

initializes local notifications on Android using the FlutterLocalNotificationsPlugin.

onDidReceiveNotification Response

HANDLE MESSAGE

custom method that takes action based on the content of the message.

When the received message is not null

If platform is Android

@itsvinayyay

**I believe!
I Code;**



@itsvinayyay

[linkedin.com/in/itsvinayyay/](https://www.linkedin.com/in/itsvinayyay/)