
Flutter Notifications: Using Flutter Local Notifications

Subtitle: Learn how to send and schedule local notifications in your Flutter apps.

Website Name: haas.dev

Website Link: <https://dev-roast-app.vercel.app>

Introduction

Notifications keep users **engaged with your app**. Flutter Local Notifications package allows you to **show, schedule, and customize notifications** easily. This guide covers setup and basic usage.

Step 1: Add Dependency

```
dependencies:  
  flutter_local_notifications: ^13.0.0
```

Run:

```
flutter pub get
```

Import in Dart:

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

Step 2: Initialize Notifications

```
final FlutterLocalNotificationsPlugin flutterLocalNotificationsPlugin =  
  FlutterLocalNotificationsPlugin();  
  
void main() async {  
  WidgetsFlutterBinding.ensureInitialized();  
  
  const AndroidInitializationSettings initializationSettingsAndroid =  
    AndroidInitializationSettings('@mipmap/ic_launcher');  
  
  const InitializationSettings initializationSettings =
```

```
InitializationSettings(android: initializationSettingsAndroid);

await flutterLocalNotificationsPlugin.initialize(initializationSettings);
runApp(MyApp());
}
```

Exercise: Initialize notifications in a new Flutter project.

Step 3: Show a Simple Notification

```
Future<void> showNotification() async {
  const AndroidNotificationDetails androidPlatformChannelSpecifics =
    AndroidNotificationDetails(
      'channel_id',
      'channel_name',
      importance: Importance.max,
      priority: Priority.high,
    );

  const NotificationDetails platformChannelSpecifics =
    NotificationDetails(android: androidPlatformChannelSpecifics);

  await flutterLocalNotificationsPlugin.show(
    0,
    'Hello!',
    'This is a simple notification',
    platformChannelSpecifics,
  );
}
```

Exercise: Trigger a notification with a button press.

Step 4: Schedule a Notification

```
Future<void> scheduleNotification() async {
  await flutterLocalNotificationsPlugin.zonedSchedule(
    1,
    'Reminder',
    'Check your tasks!',
    tz.TZDateTime.now(tz.local).add(Duration(seconds: 10)),
    const NotificationDetails(
      android: AndroidNotificationDetails('channel_id', 'channel_name'),
    ),
  );
}
```

```
),  
  androidAllowWhileIdle: true,  
  uiLocalNotificationDateInterpretation:  
    UILocalNotificationDateInterpretation.absoluteTime,  
);  
}
```

Exercise: Schedule a notification to appear 10 seconds after a button click.

Step 5: Mini Project

Build a **Flutter Reminder App**:

- Button to show immediate notification
 - Schedule notification at a future time
 - Use custom title and message
 - Bonus: Add multiple notifications with different IDs
-

Key Takeaways

- Notifications increase **user engagement and retention**
 - Use initialization, show, and schedule methods for different use cases
 - Always handle notification channels properly for Android
 - Mastering notifications adds **professional polish to your apps**
-

Skipping notifications makes your app **less interactive and engaging**, which can reduce user retention.

Visit **haas.dev** for more Flutter guides, tutorials, and complete project resources.

Website Name: haas.dev

Website Link: <https://dev-roast-app.vercel.app>
