
Flutter Bluetooth Integration: Connecting with Devices

Subtitle: Learn how to connect and communicate with Bluetooth devices in Flutter apps.

Website Name: haas.dev

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

Introduction

Bluetooth integration allows apps to **connect to devices like speakers, IoT gadgets, or wearables**. Flutter provides packages to **scan, connect, and exchange data over Bluetooth**.

Step 1: Add Dependencies

```
dependencies:  
  flutter_blue: ^1.4.0
```

Run:

```
flutter pub get
```

Import in Dart:

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

Step 2: Scan for Devices

```
FlutterBlue flutterBlue = FlutterBlue.instance;  
  
flutterBlue.startScan(timeout: Duration(seconds: 5));  
  
var subscription = flutterBlue.scanResults.listen((results) {  
  for (ScanResult r in results) {  
    print('${r.device.name} found! ${r.device.id}');  
  }  
});
```

Exercise: Print available Bluetooth devices in the console.

Step 3: Connect to a Device

```
BluetoothDevice device; // obtained from scan
await device.connect();
print('Connected to ${device.name}');
```

Exercise: Connect to a nearby device found during scan.

Step 4: Discover Services & Send Data

```
List<BluetoothService> services = await device.discoverServices();
var characteristic = services[0].characteristics[0];
await characteristic.write([0x12, 0x34]);
```

Exercise: Send a byte array to the connected Bluetooth device.

Step 5: Mini Project

Build a **Flutter Bluetooth Controller App**:

- Scan and list nearby devices
 - Connect and send test data
 - Bonus: Read data from sensors and display live updates
-

Key Takeaways

- Bluetooth allows **device-to-device communication** without internet
 - Proper scanning, connection, and service discovery is essential
 - Flutter packages simplify **integration with IoT and peripheral devices**
 - Mastering Bluetooth makes your app **hardware-interactive and real-world ready**
-

Skipping Bluetooth integration limits your app's **ability to connect with external hardware**, reducing functionality for IoT and wearable applications.

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

Website Name: haas.dev

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

