
Flutter Internationalization (i18n): Multi-Language Apps

Subtitle: Learn how to make your Flutter app support multiple languages for global users.

Website Name: haas.dev

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

Introduction

Supporting multiple languages **expands your app's reach**. Flutter provides tools to **translate text, switch languages dynamically, and handle localization**.

Step 1: Add Dependencies

```
dependencies:  
  flutter_localizations:  
    sdk: flutter  
  intl: ^0.18.1
```

Run:

```
flutter pub get
```

Import in Dart:

```
import 'package:flutter_localizations/flutter_localizations.dart';  
import 'package:intl/intl.dart';
```

Step 2: Set Up Localization Delegates

```
MaterialApp(  
  localizationsDelegates: [  
    GlobalMaterialLocalizations.delegate,  
    GlobalWidgetsLocalizations.delegate,  
    GlobalCupertinoLocalizations.delegate,  
  ],  
  supportedLocales: [  
    const Locale('en', ''),
```

```
const Locale('es', ''),  
],  
home: HomeScreen(),  
)
```

Exercise: Add English and Spanish support in your app.

Step 3: Use Translated Strings

```
String greeting = Intl.message(  
  'Hello World',  
  name: 'greeting',  
  desc: 'Greeting message',  
);
```

Exercise: Display greeting in multiple languages based on device locale.

Step 4: Create ARB Files

- lib/l10n/intl_en.arb

```
{  
  "greeting": "Hello World"  
}
```

- lib/l10n/intl_es.arb

```
{  
  "greeting": "Hola Mundo"  
}
```

Run:

```
flutter pub run intl_utils:generate
```

Exercise: Add two more ARB files for any other language.

Step 5: Mini Project

Build a **Multi-Language Flutter App**:

- Support English, Spanish, and one more language
 - Switch language dynamically in settings
 - Bonus: Localize dates, numbers, and currencies
-

Key Takeaways

- Internationalization expands your app's **global reach**
 - `intl` and ARB files simplify **translations**
 - Handling locale dynamically improves **user experience**
 - Mastering i18n is essential for **professional, multi-language apps**
-

Skipping internationalization limits your app to **single-language users**, reducing accessibility and global adoption.

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

Website Name: haas.dev

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