
Flutter File Handling: Reading & Writing Files Locally

Subtitle: Learn how to manage local files in Flutter for storing and retrieving app data.

Website Name: haas.dev

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

Introduction

Some apps need to **store large data, logs, or user-generated content** locally. Flutter allows you to **read and write files** on the device efficiently using `dart:io` and `path` utilities.

Step 1: Add Dependencies

```
dependencies:  
  path_provider: ^2.2.0
```

Run:

```
flutter pub get
```

Import in Dart:

```
import 'dart:io';  
import 'package:path_provider/path_provider.dart';
```

Step 2: Get Local Directory

```
Future<String> getLocalPath() async {  
  final directory = await getApplicationDocumentsDirectory();  
  return directory.path;  
}
```

Exercise: Print the app's local documents directory path.

Step 3: Write Data to File

```
Future<int> writeData(String fileName, String data) async {  
  final directory = await getApplicationDocumentsDirectory();  
  final file = File('${directory.path}/$fileName');
```

```
Future<Void> writeData(String filename, String data) async {
  final path = await getLocalPath();
  final file = File('$path/$filename.txt');
  await file.writeAsString(data);
}
```

Exercise: Save a simple text note to a file locally.

Step 4: Read Data from File

```
Future<String> readData(String filename) async {
  try {
    final path = await getLocalPath();
    final file = File('$path/$filename.txt');
    return await file.readAsString();
  } catch (e) {
    return 'Error reading file: $e';
  }
}
```

Exercise: Display saved file content in the app.

Step 5: Mini Project

Build a **Flutter Notes App**:

- Create, save, and read multiple notes
 - Store notes as individual text files
 - Bonus: Add search and delete functionality
-

Key Takeaways

- File handling enables **persistent storage of complex or large data**
 - Path provider gives **safe access to app-specific directories**
 - Reading and writing files is essential for **offline-first apps**
 - Mastering file handling ensures your app **can manage user-generated content**
-

Skipping file handling limits your app to **temporary or server-only data**, reducing offline capabilities and user flexibility.

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

Website Name: haas.dev

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