
Flutter Authentication: Firebase Email & Password Login

Subtitle: Learn how to implement secure user authentication in Flutter using Firebase.

Website Name: haas.dev

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

Introduction

Authentication is a core part of most apps. Firebase makes it simple to **add email/password login, signup, and logout** functionality in Flutter. This guide walks through the setup and basic usage.

Step 1: Add Firebase Auth Dependency

```
dependencies:  
  firebase_auth: ^4.6.7  
  firebase_core: ^3.27.0
```

Run:

```
flutter pub get
```

Import in Dart:

```
import 'package:firebase_auth/firebase_auth.dart';  
import 'package:firebase_core/firebase_core.dart';
```

Step 2: Initialize Firebase

```
void main() async {  
  WidgetsFlutterBinding.ensureInitialized();  
  await Firebase.initializeApp();  
  runApp(MyApp());  
}
```

Step 3: Sign Up User

```
Future<void> signUp(String email, String password) async {
  try {
    await FirebaseAuth.instance.createUserWithEmailAndPassword(
      email: email,
      password: password,
    );
    print('User registered successfully');
  } catch (e) {
    print('Error: $e');
  }
}
```

Exercise: Create a signup form to register users with email and password.

Step 4: Sign In User

```
Future<void> signIn(String email, String password) async {
  try {
    await FirebaseAuth.instance.signInWithEmailAndPassword(
      email: email,
      password: password,
    );
    print('User logged in');
  } catch (e) {
    print('Error: $e');
  }
}
```

Exercise: Create a login form and print user info after successful login.

Step 5: Sign Out User

```
Future<void> signOut() async {
  await FirebaseAuth.instance.signOut();
  print('User logged out');
}
```

Exercise: Add a logout button that returns the user to the login screen.

Step 6: Check Authentication State

```
FirebaseAuth.instance.authStateChanges().listen((User? user) {  
  if (user == null) {  
    print('User is signed out');  
  } else {  
    print('User is signed in');  
  }  
});
```

Exercise: Redirect users to Home screen if signed in, or Login screen if signed out.

Step 7: Mini Project

Build a **Flutter Firebase Auth App**:

- Signup screen with email & password
 - Login screen for existing users
 - Logout button
 - Authentication state listener to redirect users
-

Key Takeaways

- Firebase Auth simplifies **secure email/password login**
 - Always handle errors and display feedback to users
 - authStateChanges ensures **dynamic navigation based on login state**
 - Authentication is essential for **personalized and secure apps**
-

Skipping authentication makes your app **insecure and unusable for real users**, which is unacceptable for production apps.

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

Website Name: haas.dev

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