

Flutter & Dart Complete Guide

Prepared by Eng. Yousef Salah Kassem

DAFL 2
Firebase

Lecture 8



Table Of Content

- Register Authentication
- Login Authentication



Firestore Authentication

You can use Firestore Authentication to allow users to sign in to your app using one or more sign-in methods, including native providers e.g., email address and password sign-in or using federated identity providers such as Google Sign-in, Twitter and Facebook Login.

To get started with Firestore authentication, once you have connected your Flutter application with Firestore, enable authentication from the Firestore console.



Get started with Firebase Auth by adding your first sign-in method

Native providers

 Email/Password

 Phone

 Anonymous

Additional providers

 Google

 Game Center

 Microsoft

 Facebook

 Apple

 Twitter

 Play Games

 GitHub

 Yahoo

Custom providers

 OpenID Connect

 SAML

Email/password authentication

Registration

If you have set field validations on the form e.g. confirm password etc, then the following registration logic will be executed when the `create account` button is clicked, if there are no form validation errors. We're also catching other auth errors which will be displayed in a toast message for the user. If the registration is successful, then we'll route to the next page.



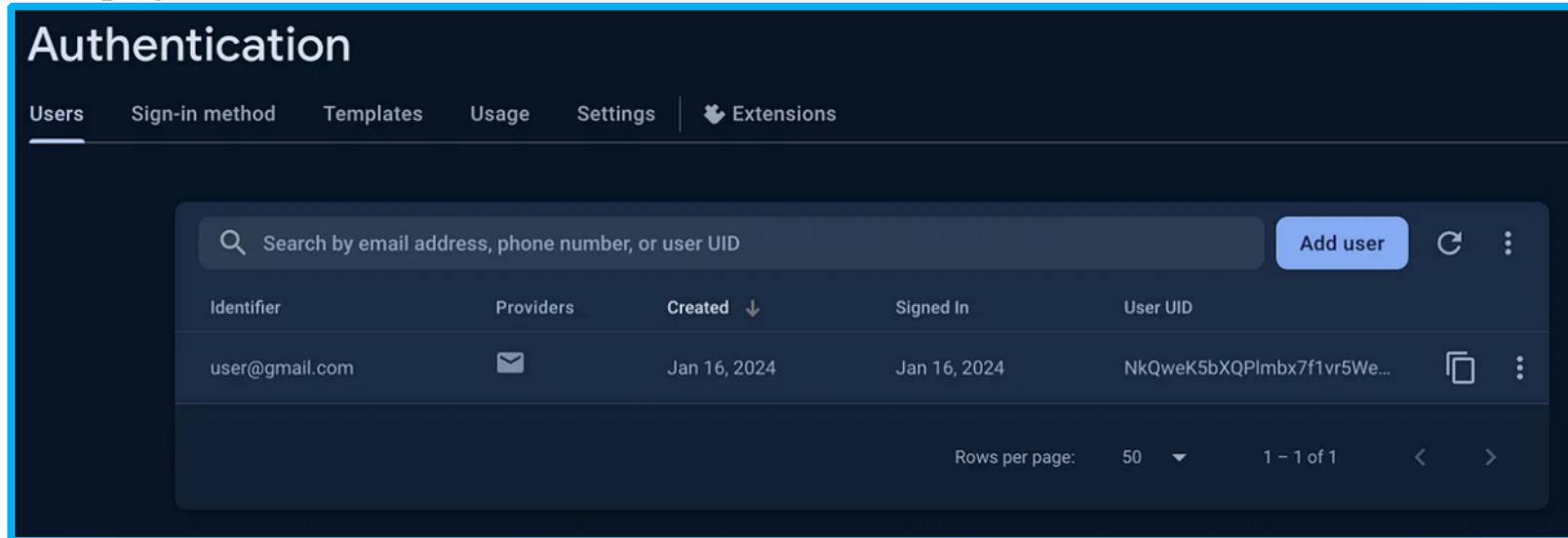
```

onTap: () async {
  String message = '';
  if (_form.currentState!.validate()) {
    try {
      await _firebaseAuth.createUserWithEmailAndPassword( // instantiated ear
        email: email.trim(),
        password: password.trim(),
      );
      Future.delayed(const Duration(seconds: 3), () {
        Navigator.of(context, rootNavigator: true).push(
          MaterialPageRoute(
            builder: (_) => const Feed(),
          ),
        );
      });
    } on FirebaseAuthException catch (e) {
      if (e.code == 'weak-password') {
        message = 'The password provided is too weak.';
      } else if (e.code == 'email-already-in-use') {
        message = 'An account already exists with that email.';
      }
      Fluttertoast.showToast(
        msg: message,
        toastLength: Toast.LENGTH_LONG,
        gravity: ToastGravity.SNACKBAR,
        backgroundColor: Colors.black54,
        textColor: Colors.white,
        fontSize: 14.0,
      );
    } catch (e) {
      Fluttertoast.showToast(
        msg: "Failed: $e",
        toastLength: Toast.LENGTH_LONG,
        gravity: ToastGravity.SNACKBAR,
        backgroundColor: Colors.black54,
        textColor: Colors.white,
        fontSize: 14.0,
      );
    }
  }
},

```

Firestore Authentication

That is pretty much the registration logic. If there are no errors, then the user will be created in the Firebase console and the next screen will be displayed.



Firestore Authentication

Login

Only users who have been registered will be able to login the application. So to login successfully, ensure you have the credentials of a user already in the database. The code for logging in is similar to that of registering. The difference is in the use of the `signInWithEmailAndPassword` as opposed to `createUserWithEmailAndPassword`



```
onTap: () async {
  String message = '';
  if (_form.currentState!.validate()) {
    try {
      await FirebaseAuth.instance.signInWithEmailAndPassword(
        email: email,
        password: password,
      );
      Future.delayed(const Duration(seconds: 3), () {
        print('success');
        Navigator.of(context, rootNavigator: true).push(
          MaterialPageRoute(
            builder: (_) => const Feed(),
          ),
        );
      });
    } on FirebaseAuthException catch (e) {
      if (e.code == 'INVALID_LOGIN_CREDENTIALS') {
        message = 'Invalid login credentials.';
      } else {
        message = e.code;
      }
      Fluttertoast.showToast(
        msg: message,
        toastLength: Toast.LENGTH_LONG,
        gravity: ToastGravity.SNACKBAR,
        backgroundColor: Colors.black54,
        textColor: Colors.white,
        fontSize: 14.0,
      );
    }
  }
},
```

Firestore Authentication

I noted something interesting while implementing this bit: With logging in, I was only able to get one `FirebaseAuthException` error i.e., `INVALID_LOGIN_CREDENTIALS`. On research, I saw that this is probably due to [email enumeration](#). Email enumeration is a type of brute-force attack in which a malicious actor attempts to guess or confirm users in a system by passing an email address to the API and checking the response. So as at 15th September 2023, the email enumeration protection feature has been enabled by default on Firestore. With this feature, the platform will return information that can't be used in an email enumeration attack. You can choose to [disable email enumeration protection](#) so that the API reverts to its previous behaviour. However, in doing that, your project/users become susceptible to the risks of an email enumeration attack.



Firestore Authentication

Ps. Once your user is registered or logged in, you might want to keep signed in until they choose to log out. You can use local storage to save the user in the app e.g., SharedPreferences, SQLite or Hive.



Firestore Authentication

Logging out

To log out a user, simply call the `signOut()` method from the log out button, icon or menu item. *(This will also apply for Google authentication signing out)*



```
IconButton(  
  onPressed: () async {  
    try {  
      await FirebaseAuth.instance.signOut();  
      Future.delayed(const Duration(seconds: 3), () {  
        Navigator.of(context, rootNavigator: true).push(  
          MaterialPageRoute(  
            builder: (_) => const LoginScreen(),  
          ),  
        );  
      });  
    } on Exception catch (e) {  
      Fluttertoast.showToast(  
        msg: e.toString(),  
        toastLength: Toast.LENGTH_LONG,  
        gravity: ToastGravity.SNACKBAR,  
        backgroundColor: Colors.black54,  
        textColor: Colors.white,  
        fontSize: 14.0,  
      );  
    }  
  },  
  icon: const Icon(CupertinoIcons.square_arrow_right),  
),
```

Create Account

5:21



NOTELY

Create a free account

Join Notely for free. Create and share unlimited notes with your friends.

Full Name

Salman Khan

Email Address

mesalmanwap@gmail.com

Password

#####

Create Account

[Already have an account?](#)

Login

5:21



NOTELY

Login

Welcome back to Notely

Email Address

mesalmanwap@gmail.com

Password

#####

[Forgot Password ?](#)

Login

[Don't have an account?](#)

Forgot Password

5:21



NOTELY

Forgot Password

We will send mail to reset your
password

Email Address

mesalmanwap@gmail.com

Reset

[Back to login](#)

[https://www.figma.com/design/VYJ3Mk36c8C7SjN076yoc3/Note-Taking-Mobile-App-\(Community\)?node-id=343-116&p=f&t=dAOt1bOa93mbgoeW-0](https://www.figma.com/design/VYJ3Mk36c8C7SjN076yoc3/Note-Taking-Mobile-App-(Community)?node-id=343-116&p=f&t=dAOt1bOa93mbgoeW-0)

Live Coding

What is Next ?



Thank You