

Mobile Application Development

Produced
by

David Drohan (ddrohan@wit.ie)

Department of Computing & Mathematics
Waterford Institute of Technology

<http://www.wit.ie>



Waterford Institute of Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE





Android & Firebase

Firebase Integration





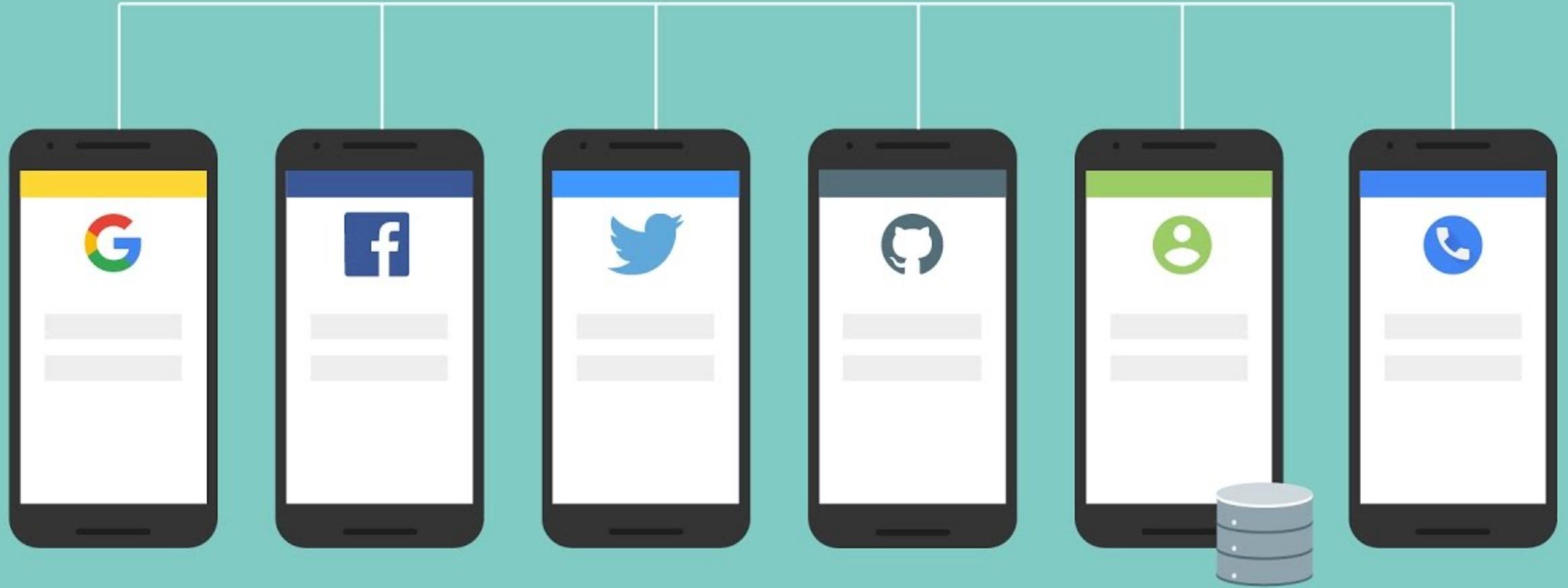
Agenda

- ❑ Firebase history
- ❑ The all new Firebase
- ❑ Real-time database
- ❑ Firestore
- ❑ Authentication
- ❑ Storage
- ❑ Remote config
- ❑ Hosting
- ❑ Crash reporting
- ❑ Test lab
- ❑ Firebase cloud messaging
- ❑ Dynamic links
- ❑ App indexing
- ❑ Analytics
- ❑ CoffeeMate Highlights & Demos along the way...



Agenda

- ❑ Firebase history
- ❑ The all new Firebase
- ❑ Real-time database
- ❑ Firestore
- ❑ **Authentication**
- ❑ Storage
- ❑ Remote config
- ❑ Hosting
- ❑ Crash reporting
- ❑ Test lab
- ❑ Firebase cloud messaging
- ❑ Dynamic links
- ❑ App indexing
- ❑ Analytics
- ❑ CoffeeMate Highlights & Demos along the way...

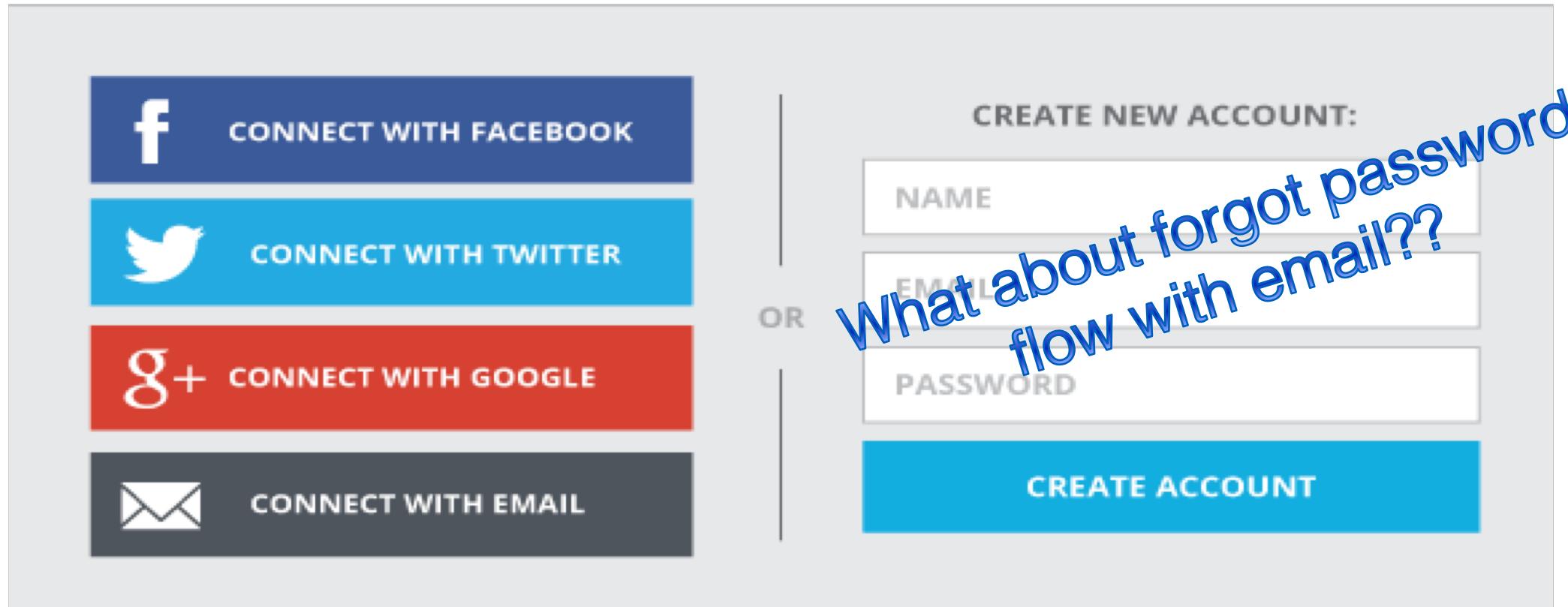


All goodness bundled as one...
Authentication



Firebase Authentication *

A type of screen present in almost all apps these days...



How long would it take you to develop this???



Firebase Authentication

- ☐ Integrate easily with popular identity providers like google, twitter, facebook and more





Firebase Authentication

- ❑ Of course you will need to register your app with individual service providers.
- ❑ Minimal client side handling, integrates seamlessly with firebase
- ❑ Ready made ‘forgot password’ flow with customizable email template



Firebase Authentication



Quick DEMO...



CoffeeMate.9.0

**Setup
&
Code
Highlights**



1. Setup your Sign-In Method *

The screenshot shows the Firebase console interface for the 'Authentication' section. The left sidebar includes 'Project Overview', 'Develop' (with 'Authentication' highlighted), 'Database', 'Storage', 'Hosting', 'Functions', and 'ML Kit'. Below these are 'Quality' (Crashlytics, Performance, Test Lab) and 'Analytics' (Dashboard, Events, Conversions, A...). At the bottom, there's a 'Spark' plan (Free \$0/month) and an 'Upgrade' button. The main content area is titled 'Authentication' and shows tabs for 'Users', 'Sign-in method' (which is selected and highlighted with a red box), 'Templates', and 'Usage'. A search bar at the top of the main content area allows searching by email address, phone number, or user UID. Below the search bar is a table header with columns: Identifier, Providers, Created, Signed In, and User UID. A large call-to-action button at the bottom right says 'Set up sign-in method'.



1. Setup your Sign-In Method *

The screenshot shows the Firebase Authentication screen for the 'CoffeeMate FullFat Server' project. The 'Sign-in method' tab is selected. A red box highlights the 'Google' provider, which is currently disabled. Other providers listed include Email/Password, Phone, Play Games, Facebook, Twitter, GitHub, and Anonymous, all of which are also disabled.

Provider	Status
Email/Password	Disabled
Phone	Disabled
Google	Disabled
Play Games	Disabled
Facebook	Disabled
Twitter	Disabled
Github	Disabled
Anonymous	Disabled



1. Setup your Sign-In Method *

The screenshot shows the Firebase console interface for setting up authentication methods. On the left, the navigation bar includes 'Project Overview', 'Develop' (with 'Authentication' selected), 'Database', 'Storage', 'Hosting', 'Functions', and 'ML Kit'. Under 'Quality', it lists 'Crashlytics, Performance, Test Lab'. Under 'Analytics', it lists 'Dashboard, Events, Conversions, A...'. At the bottom, there's a 'Spark' section with 'Free \$0/month' and an 'Upgrade' button.

The main content area is titled 'Authentication' under 'CoffeeMate FullFat Server'. It shows a 'Phone' sign-in method as 'Disabled'. Below that is a 'Google' sign-in method. An 'Enable' toggle switch is highlighted with a red rectangle. A note below the switch states: "Google sign-in is automatically configured on your connected iOS and web apps. To set up Google sign-in for your Android apps, you need to add the [SHA1 fingerprint](#) for each app on your [Project Settings](#)."

A modal window is open, prompting to "Update the [project-level setting](#) below to continue". It contains fields for "Project public-facing name" (set to "project-372308519667") and "Project support email" (set to "Not configured"). A tooltip for the project name explains: "This will be the name presented to users when they are shown any public instances of your project. For example, this will be the name displayed on emails that your users receive after creating an account with your app."

At the bottom of the modal, there's a note: "Please select an email address" with a red exclamation mark icon.



1. Setup your Sign-In Method *

The screenshot shows the Firebase console interface for a project titled "CoffeeMate FullFat Server". The left sidebar includes sections for Project Overview, Develop (Authentication, Database, Storage, Hosting, Functions, ML Kit), Quality (Crashlytics, Performance, Test Lab), and Analytics (Dashboard, Events, Conversions, A...). The main content area is titled "Authentication" and has an "Enable" toggle switch turned on. A note states: "Google sign-in is automatically configured on your connected iOS and web apps. To set up Google sign-in for your Android apps, you need to add the [SHA1 fingerprint](#) for each app on your [Project Settings](#)." Below this, a modal dialog box is open, prompting to "Update the [project-level setting](#) below to continue". It contains fields for "Project public-facing name" (set to "CoffeeMate Full Fat") and "Project support email" (set to "daveydrohan@gmail.com"). At the bottom right of the modal is a red-bordered "Save" button.



1. Setup your Sign-In Method *

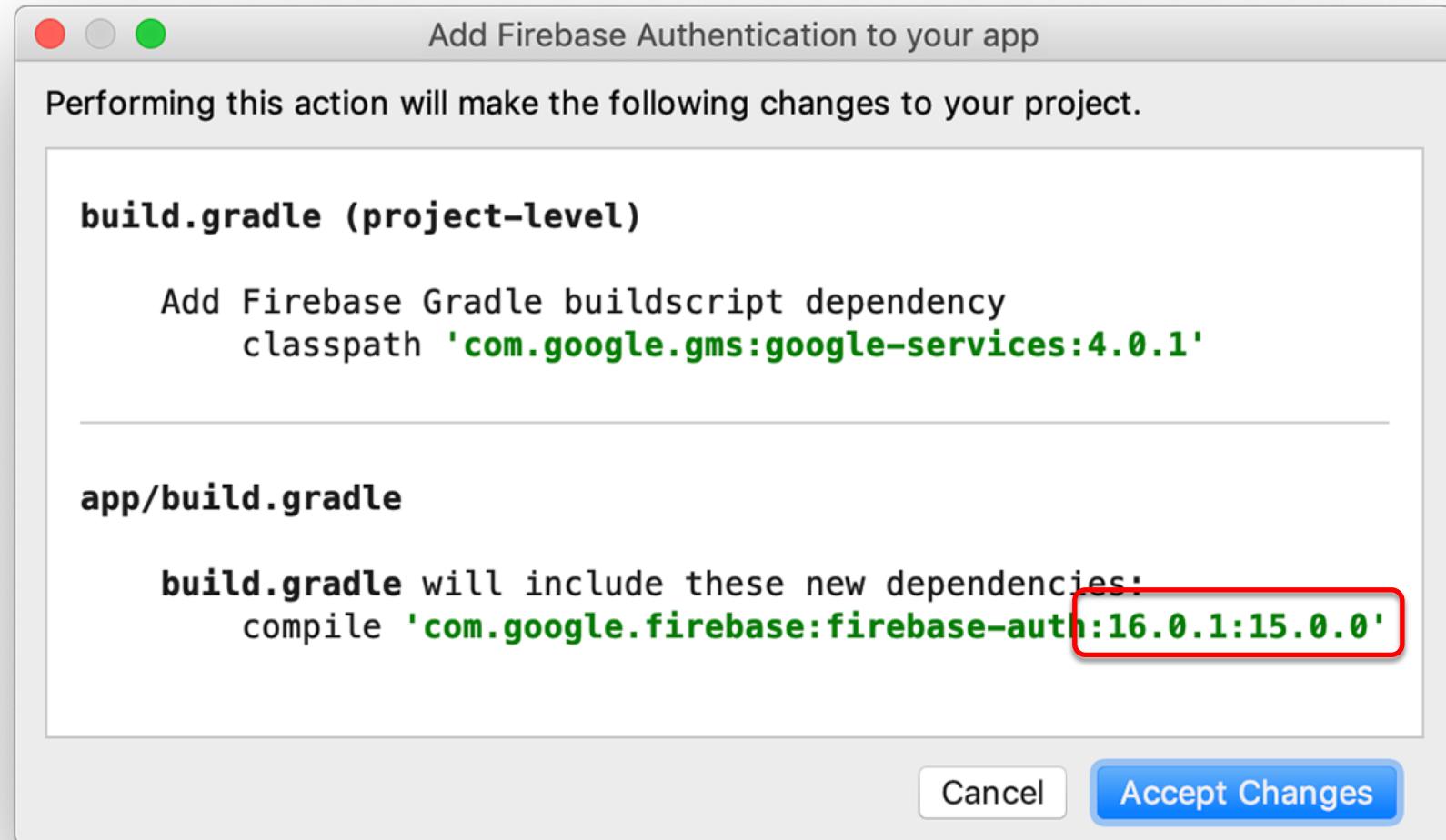
The screenshot shows the Firebase Authentication screen in the Firebase console. The left sidebar is dark blue with white text and icons. The main area has a light blue header bar with the project name and tabs for 'Sign-in methods' (which is selected) and 'Email/Password'. Below this is a table titled 'Sign-in providers'.

Provider	Status
Email/Password	Enabled
Phone	Disabled
Google	Enabled
Play Games	Disabled
Facebook	Disabled
Twitter	Disabled
Github	Disabled
Anonymous	Disabled

At the bottom of the main area, there is a link 'Authorised domains' with a question mark icon.



1. Still an issue (as of December 2018) *





1. Still an issue (as of December 2018) *

```
implementation 'com.google.firebaseio:firebase-core:16.0.1'  
implementation 'com.google.firebaseio:firebase-auth:16.0.1'
```

```
}
```

```
dependencies {  
    classpath 'com.android.tools.build:gradle:3.2.1'  
    classpath 'com.google.gms:google-services:3.2.0'
```



2. Introduce Authentication Flow *

```
private void firebaseAuthWithGoogle(GoogleSignInAccount acct) {
    Log.v(TAG, "firebaseAuthWithGoogle:" + acct.getEmail());

    AuthCredential credential = GoogleAuthProvider.getCredential(acct.getIdToken(), null);
    app.mFirebaseAuth.signInWithCredential(credential)
        .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                Log.v(TAG, "signInWithCredential:onComplete:" + task.isSuccessful());
                validateFirebaseUser();
                // If sign in fails, display a message to the user. If sign in succeeds
                // the auth state listener will be notified and logic to handle the
                // signed in user can be handled in the listener.
                if (!task.isSuccessful()) {
                    Log.v(TAG, "signInWithCredential", task.getException());
                    Toast.makeText(Login.this, "Authentication failed.",
                        Toast.LENGTH_SHORT).show();
                }
            }
        });
}
```



2. Introduce Authentication Flow – Check for User *

```
public void validateFirebaseUser() {  
    if(app.mFirebaseUser == null)  
        app.mFirebaseUser = FirebaseAuth.getInstance().getCurrentUser();  
  
    Log.v(TAG, "checkUser ID == " + app.mFirebaseUser.getUid());  
    app.mFirebaseManager.m FirebaseDatabase.child("users")  
        .child(app.mFirebaseUser.getUid()).addValueEventListener(  
    new ValueEventListener() {  
        @Override  
        public void onDataChange(DataSnapshot dataSnapshot) {...}  
  
        @Override  
        public void onCancelled(DatabaseError databaseError) {...}  
    }  
);  
}
```



2. Introduce Authentication Flow – Check for User *

```
app.mFirebaseManager.m FirebaseDatabase.child("users")
    .child(app.mFirebaseUser.getUid()).addListenerForSingleValueEvent(
        new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {...}

            @Override
            public void onCancelled(DatabaseError databaseError) {...}
        }
);
```



2. Introduce Authentication Flow – Success *

```
app.mFirebaseManager.m FirebaseDatabase.child("users")
    .child(app.mFirebaseUser.getUid()).addSingleValueEvent(
new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        Log.v(TAG, "mFirebaseListener.onSuccess " + dataSnapshot);
        if(dataSnapshot.exists()){
            Log.v(TAG, "User found : " + app.mFirebaseUser.getEmail());
        }
        else{
            Log.v(TAG, "User not found, Creating User on Firebase");
            User newUser = new User(app.mFirebaseUser.getUid(),
                app.mFirebaseUser.getDisplayName(),
                app.mFirebaseUser.getEmail(), null);
            app.mFirebaseManager.m FirebaseDatabase.child("users")
                .child(app.mFirebaseUser.getUid())
                .setValue(newUser);
        }
        app.mFirebaseManager.m FirebaseUserId = app.mFirebaseUser.getUid();
        startHomeScreen();
    }
    @Override
    public void onCancelled(DatabaseError databaseError) { }
```



2. Introduce Authentication Flow – Failure *

```
app.mFirebaseManager.m FirebaseDatabase.child("users")
    .child(app.mFirebaseUser.getUid()).addSingleValueEvent(
new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {...}

    @Override
    public void onCancelled(DatabaseError databaseError) {
        Log.v(TAG, "mFirebaseListener.onFailure " + databaseError);
        Log.v(TAG, "Unable to Validate Existing Firebase User: "
            + app.mFirebaseUser.getUid());
        Toast.makeText(Login.this,"Unable to Validate Existing Firebase User:",
                    Toast.LENGTH_LONG).show();
    }
};
```



Firebase Console – Authenticated Users *

The image shows the Firebase Authentication console for the 'CoffeeMate FullFat Server' project. The 'Users' tab is selected, highlighted with a red box. A single user record is listed, also highlighted with a red box. The user's email is daveydrohan@gmail.com, they signed up via Google on 11 Dec 2020, and their User UID is R9QfVfQgs3gJpwd5sebmEy... .

Identifier	Provider	Created	Signed In	User UID
daveydrohan@gmail.com	G	11 Dec 20...	11 Dec 20...	R9QfVfQgs3gJpwd5sebmEy...

Below the table, there are navigation controls: 'Rows per page: 50', '1-1 of 1', and arrows for navigation.

To the right of the console screenshot is a mobile phone displaying the CoffeeMate app. The app's sidebar menu includes Home, Add a Coffee, Search, View Favourites, Take a Photo, Share, and View on Map. The main screen shows a list of coffee items with prices and ratings, each marked with a red 'X'.

A vertical toolbar on the far right of the phone screen contains icons for power, volume, camera, search, and other settings.



Firebase Console – Authenticated Users *

The screenshot shows the Firebase Authentication console for the "CoffeeMate FullFat Server" project. The left sidebar includes links for Project Overview, Authentication (which is selected), Database, Storage, Hosting, Functions, ML Kit, Quality (Crashlytics, Performance, Test Lab), Analytics (Dashboard, Events, Conversions, A...), and Spark (Free \$0/month, Upgrade). The main "Authentication" page has tabs for Users, Sign-in method, Templates, and Usage. It features a search bar and a table with columns for Identifier, Providers, Created, Signed In, and User UID. The table displays three users:

Identifier	Providers	Created	Signed In	User UID
daveydrohan@gmail.com	G	11 Dec 20...	11 Dec 20...	R9QfVfQgs3gJpwd5sebmEy...
noahldrohan@gmail.com	G	11 Dec 20...	11 Dec 20...	dyl0cAMbtuVXdED5ZKIKek6...
joshuajdrohan@gmail.com	G	11 Dec 20...	11 Dec 20...	t3KpD6ilhAcKp1nNbKic2rDZ...

A vertical toolbar on the right side of the mobile device screen provides navigation icons for power, volume, camera, search, and more.

The mobile application interface shows a user profile for "Joshua Everett Drohan" with the email "joshuajdrohan@gmail.com". The navigation menu includes Home, Add a Coffee, Search, View Favourites, Take a Photo, Share, and View on Map. A vertical toolbar on the right side of the mobile device screen provides navigation icons for power, volume, camera, search, and more.



Firebase Console – Authenticated Users *

The screenshot shows the Firebase Authentication console under the 'Users' tab. It lists three users with their email addresses, provider (Google), creation date (11 Dec 2020), sign-in date (11 Dec 2020), and User UID. The User UIDs are partially visible: R9QfVfQgs3gJpwd5sebmEy... for David, dyl0cAMbtuVXdED5ZKIKek6... for Noah, and t3KpD6ilhAckp1nNbKic2rDZ... for Joshua.

Identifier	Providers	Created	Signed In	User UID
daveydrohan@gmail.com	G	11 Dec 20...	11 Dec 20...	R9QfVfQgs3gJpwd5sebmEy...
noahldrohan@gmail.com	G	11 Dec 20...	11 Dec 20...	dyl0cAMbtuVXdED5ZKIKek6...
joshuajdrohan@gmail.com	G	11 Dec 20...	11 Dec 20...	t3KpD6ilhAckp1nNbKic2rDZ...

The screenshot shows a mobile application's sign-in screen. It displays three Google accounts: David Drohan, Noah Drohan, and Joshua Everett Drohan. Each account has a profile picture and an email address below it. A red box highlights all three accounts. At the bottom, there is a link to "Use another account".



Firebase Console – User Data in db *

The screenshot shows the Firebase Database console for the project "CoffeeMate FullFat Server". The left sidebar has a red box around the "Database" item under the "Develop" section. The main area shows the Realtime Database structure at `https://coffeemate-fullfat-server-46ff4.firebaseio.com/`. A red box highlights a user node under the "users" reference, containing the following data:

```
users
  R9QfVfQgs3gJpwd5sebmEy34dmB3
    userEmail: "daveydrohan@gmail.com"
    userId: "R9QfVfQgs3gJpwd5sebmEy34dmB3"
    userName: "David Drohan"
```



Firebase Console – User Data in db *

The screenshot shows the Firebase Database console for the project "CoffeeMate FullFat Server". The left sidebar is titled "Develop" and includes links for Authentication, **Database**, Storage, Hosting, Functions, and ML Kit. The "Database" link is highlighted with a red box. The main area displays the Realtime Database structure under "coffeemate-fullfat-server-46ff4". A specific node under "users" is highlighted with a red box and contains the following data:

```
users
  R9O...: ...
  dyI0cAMbtuVXdED5ZKIKeK6hqhu1:
    userEmail: "noahldrohan@gmail.com"
    userId: "dyI0cAMbtuVXdED5ZKIKeK6hqhu1"
    userName: "Noah Drohan"
  t3K...: ...
```



Firebase Console – User Data in db *

The screenshot shows the Firebase Database console for the project "CoffeeMate FullFat Server". The left sidebar has a red box around the "Database" item under the "Develop" section. The main area shows the database structure:

```
coffeemate-fullfat-server-46ff4
  +-- coffees
  +-- user-coffees
  +-- users
      +-- R9Q8...: 4dmB3
      +-- dylo...: hqhu1
      +-- t3...: 672
```

A large red box highlights the "users" node and its three child nodes.



Some important points though...

- ❑ Do not think RDBMS, think JSON. How data should be structured is very important.
- ❑ Firebase has a recycler view, that integrates with real time database smoothly without any listeners. (FirebaseUI)
- ❑ Test lab which is available in paid plan (Blaze), is an amazing feature for testing your app on different real and virtual devices (next section)
- ❑ Set developer mode to true when testing Remote Config (next section).



References & Links

- ❑ [Presentation by Kaushal Dhruw & Shakti Moyal 2016](#)
- ❑ <https://firebase.google.com>
- ❑ Demo app available at <https://goo.gl/WBP5fR>



Questions?

