Updating to the Latest Firebase Package Versions

In this module, we used `cloud_firestore: 0.13.5` etc.

If you want to use the latest versions, you can follow the official migration guide to update your code: https://firebase.flutter.dev/docs/migration

---

Attached you find the updated project code!

Especially make sure, that you remove and re-add the Android and iOS App configuration files and file entries - see: https://firebase.flutter.dev/docs/migration#3-platform-setup

Then, update your dependencies and code like this:

Add the following **NEW dependency** to the `pubspec.yaml` file:

```
firebase_core: "^0.5.0"
```

Update the Firebase Auth and Cloud Firestore dependencies in your `pubspec.yaml` file - e.g. to:

```
firebase_auth: "^0.18.0+1"
cloud_firestore: "^0.14.0+2"
```

Now, let's dive into the code.

First, you have to call `Firebase.initializeApp();` early in your app.

You can do this in `main.dart`:

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


...


final Future<FirebaseApp> _initialization = Firebase.initializeApp();
    return FutureBuilder(
      // Initialize FlutterFire:
      future: _initialization,
      builder: (context, appSnapshot) {
        return MaterialApp(
          ...
          home: appSnapshot.connectionState != ConnectionState.done ? SplashScreen() : StreamBuilder(stream: FirebaseAuth.instance.authStateChanges(), ...
```

`initializeApp()` returns a `Future`, hence we use a `FutureBuilder` to either show the `SplashScreen()` if we're still waiting for app initialization or to show or regular `StreamBuilder`-based output if the app was initialized.

Also note that - as an additional change - `onAuthStateChanged` was renamed to `authStateChanges` and is **called as a method** now!

That's it for the `main.dart` file.

In the `auth_screen.dart` file, note that the `AuthResult` type was renamed to `UserCredential`.

Hence

```
AuthResult authResult;
```

becomes

```
UserCredential authResult;
```

In addition - in this file and in all other files - `Firestore` (e.g. in `Firestore.instance`) should be renamed to `FirebaseFirestore` (i.e. we would use `FirebaseFirestore.instance`).

In addition, the `document()` method on a `collection(...)` was renamed to `doc()`, `setData(...)` was renamed to `set(...)`.

You can still use the old names but it's better if you update your code - e.g.

```
 await Firestore.instance
  .collection('users')
  .document(authResult.user.uid)
  .setData({
    'username': username,
    'email': email,
    'image_url': url,
  });
```

becomes

```
  await FirebaseFirestore.instance
  .collection('users')
  .doc(authResult.user.uid)
  .set({
    'username': username,
    'email': email,
    'image_url': url,
  });
```

In `messages.dart` and any other file where we rely on the `currentUser()`, `currentUser()` no longer returns a `Future` but instead acts as a property which immediately returns the current user object - synchronously!

Hence

```
return FutureBuilder(
  future: FirebaseAuth.instance.currentUser(),
    builder: (ctx, futureSnapshot) {
      if (futureSnapshot.connectionState == ConnectionState.waiting) {
        return Center(
          child: CircularProgressIndicator(),
        );
      }
      return StreamBuilder(
        stream: Firestore.instance,
        builder: ...
```

becomes

```
final user = FirebaseAuth.instance.currentUser;
return StreamBuilder(
  stream: FirebaseFirestore.instance,
  builder: ...
```

In addition - since user is no longer a Future:

```
chatDocs[index]['userId'] == futureSnapshot.data.uid
```

becomes

```
chatDocs[index]['userId'] == user.uid;
```

Last but not least, when accessing the data provided by a Firebase query - like in `new_message.dart`, the syntax also changed.

Instead of

```
'username': userData['username']
```

you would now use the extra `data()` method to extract the data:

```
'username': userData.data()['username']
```

This means, that the code in `messages.dart` also needs an adjustment:

```
 final chatDocs = chatSnapshot.data.documents;
  return ListView.builder(
    reverse: true,
    itemCount: chatDocs.length,
    itemBuilder: (ctx, index) => MessageBubble(
      chatDocs[index]['text'],
      chatDocs[index]['username'],
      chatDocs[index]['userImage'],
      chatDocs[index]['userId'] == user.uid,
      key: ValueKey(chatDocs[index].documentID),
    ),
  );
```

becomes

```
  final chatDocs = chatSnapshot.data.docs;
    return ListView.builder(
      reverse: true,
      itemCount: chatDocs.length,
      itemBuilder: (ctx, index) => MessageBubble(
          chatDocs[index].data()['text'],
          chatDocs[index].data()['username'],
          chatDocs[index].data()['userImage'],
          chatDocs[index].data()['userId'] == user.uid,
          key: ValueKey(chatDocs[index].id),
        ),
    );
```

Note that `documents` was changed to `docs`, the `data()` method was added and `documentID` was renamed to just `id`.

The code for handling push notifications also has to be adjusted slightly - to be precise, this code in `_ChatScreenState`:

```
void initState() {
    super.initState();
    final fbm = FirebaseMessaging();
    fbm.requestNotificationPermissions();
    fbm.configure(onMessage: (msg) {
      print(msg);
      return;
    }, onLaunch: (msg) {
      print(msg);
      return;
    }, onResume: (msg) {
      print(msg);
      return;
    });
    fbm.subscribeToTopic('chat');
}
```

becomes

```
void initState() {
    super.initState();
    final fbm = FirebaseMessaging.instance;
    fbm.requestPermission();
    FirebaseMessaging.onMessage.listen((message) {
      print(message);
      return;
    });
    FirebaseMessaging.onMessageOpenedApp.listen((message) {
       print(message);
      return;
    });
    fbm.subscribeToTopic('chat');
}
```

In that adjustment, instantiating `FirebaseMessaging()` was replaced with calling `instance()` on `FirebaseMessaging` instead. In addition, calling `configure()` with `onMessage`, `onLaunch` and `onResume` was replaced with calling `FirebaseMessaging.onMessage.listen(...)` (instead of `onMessage`) and calling `FirebaseMessaging.onMessageOpenedApp.listen(...)` (instead of `onLaunch` and `onResume`).

To finish it up, in `auth_screen.dart` :

```
await ref.putFile(image).onComplete;
```

becomes just

```
await ref.putFile(image);
```

That's it. Again, you find the **updated code snapshot attached to this lecture!**