**A Note About Token Expiration**

The token we're managing in our context and which we're storing on our device**could expire**. To be precise, it will **depend on your backend**, whether it expires or not.

In Firebase' case, there indeed is a **1h timer** on the token. Therefore, the token can still used after 1h but if you're sending it to Firebase, to access some protected resources, after that hour, it won't work anymore. Firebase will deny access.

Therefore, you might want to consider doing at **least one of two things** in your app code:

1. **Automatically log the user out** after 1h (to avoid that the user thinks he or she is logged in)
2. **Refresh the token** and get a new auth token

1. can be achieved with help of setTimeout(). You can set a timer which runs in the background and logs the user out after 1h. Of course you should set that timer to a correct duration. If you just got a new token, the duration will be 1h. If you logged a user in because the token was stored on the device, the remaining duration is likely less than 1 hour. You should then calculate the remaining duration by subtracting the current time from the expiration time determine when the token was first received. Therefore, this expiration time should also be derived and stored, whenever a new token is received.

2. can be achieved with help of a specific auth API endpoint provided by Firebase - this endpoint: <https://firebase.google.com/docs/reference/rest/auth#section-refresh-token>

Whenever you get an auth token (i.e., after logging in or creating a new user), you also get a refresh token (see the official documentation for [logging in](https://firebase.google.com/docs/reference/rest/auth#section-sign-in-email-password) and [signing up](https://firebase.google.com/docs/reference/rest/auth#section-create-email-password). The refreshToken field in the responses carries a token that can be sent to the refresh token API endpoint (<https://firebase.google.com/docs/reference/rest/auth#section-refresh-token>) to get a new auth token. For this to work, the refresh token should of course also be stored (in Context and on the device).

You could set a timer to refresh the token periodically, or you refresh it whenever the auth token expired.

If you have a backend where tokens don't expire, the above steps of course won't apply to you.

Alternatively, many third-party services (like Firebase) offer official SDKs which handle token management (and refreshing the token) for you.