

React Native Firebase Authentication

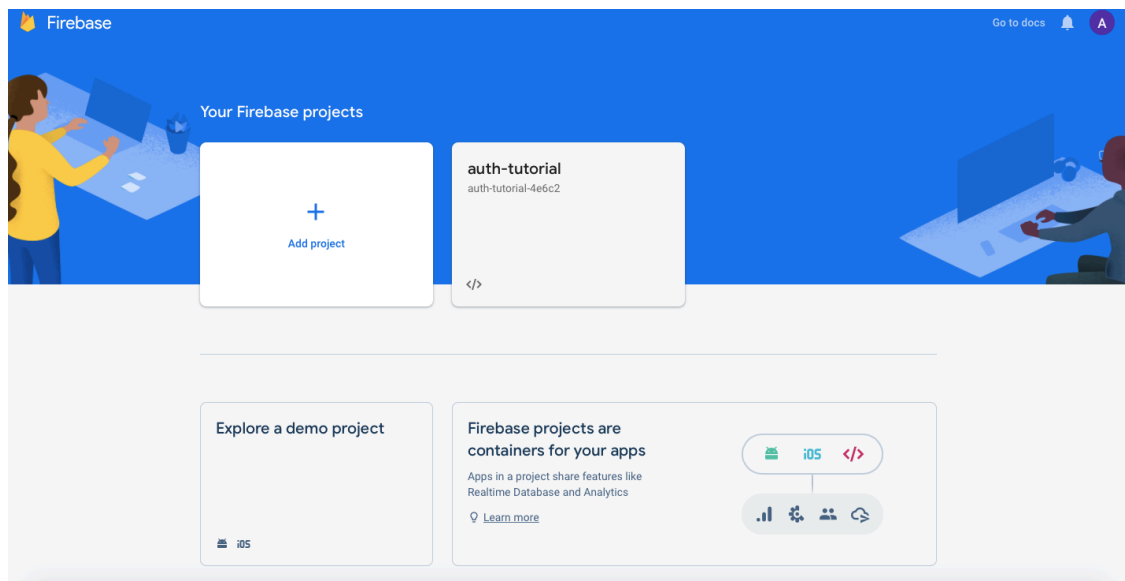
What is Firebase?

Firebase is a Cloud-hosted, NoSQL database which lets you store and synchronize data in real-time among users and is built on the Google infrastructure. In addition to standard NoSQL database functionality, Firebase includes analytics, authentication, performance monitoring, messaging, crash reporting etc. Because it is a Google product, there is also integration into a lot of other products such as the Play Store, Data Studio.

How to use Firebase?

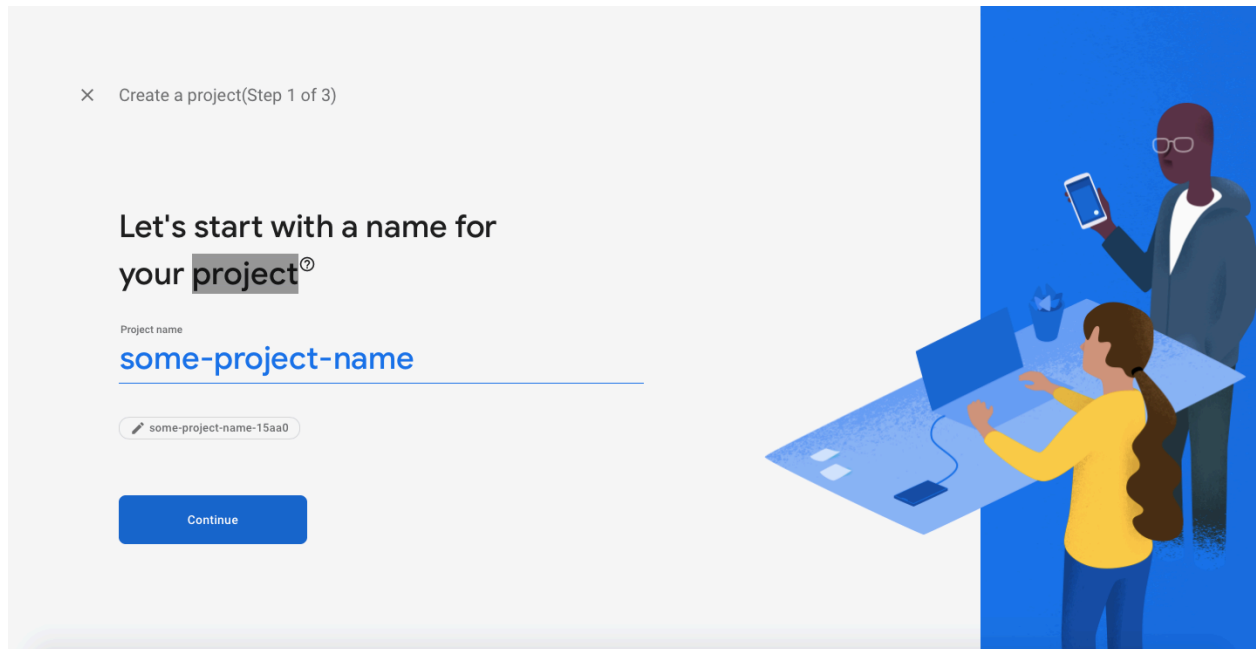
To use the authentication feature of Firebase follow the next steps where the new project is created on already signed in account:

1. As soon as you sign in to your google account, you are presented with page where there is an option "Add project". Click on it and move to the next page.

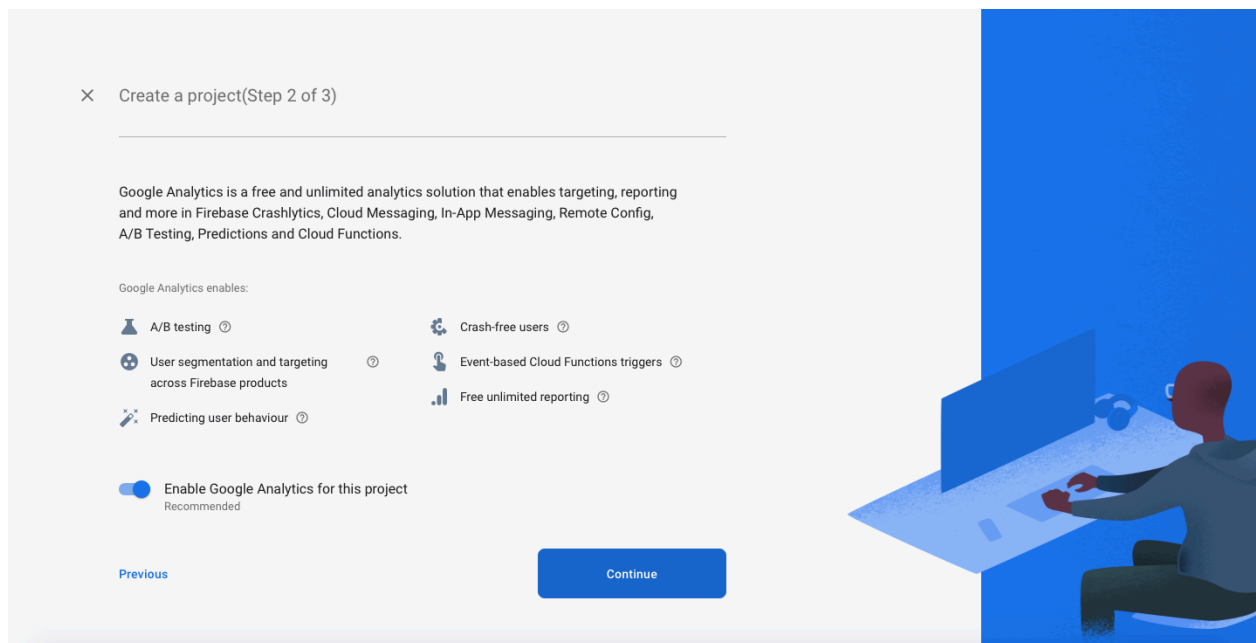


Step 1

2. In second page, write the name of project that you are creating press Continue (Step 2(a)). In following page review the information about Google Analytics and press Continue (Step 2(b)).

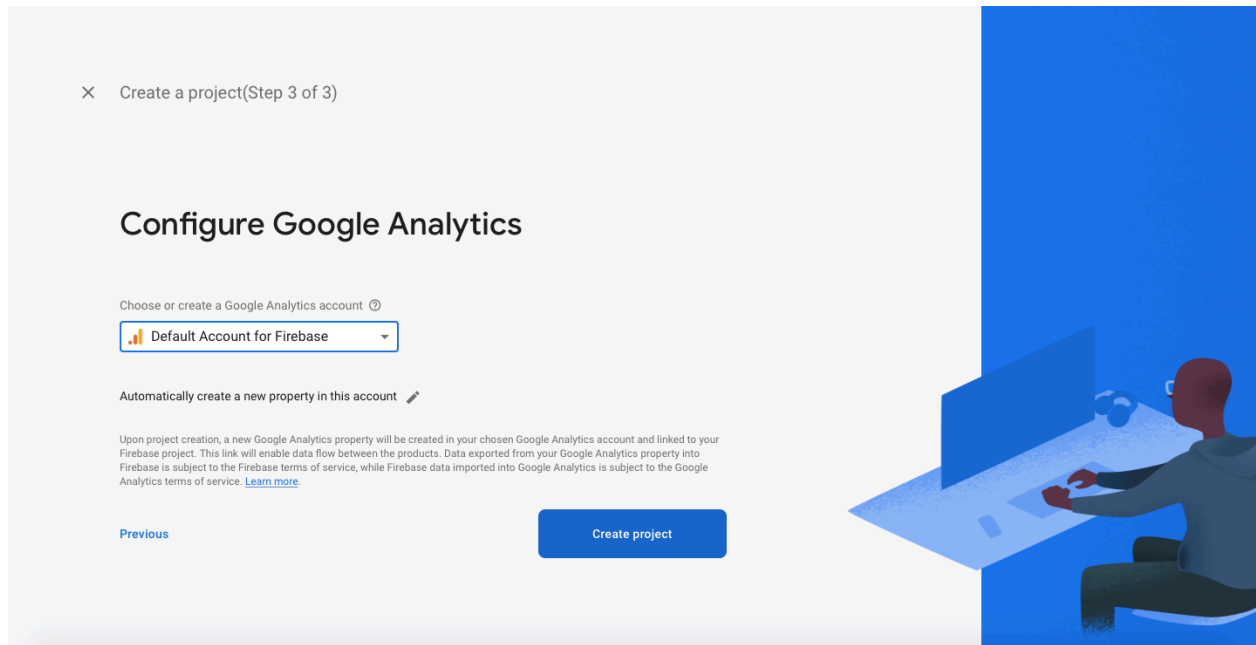


Step 2 (a)

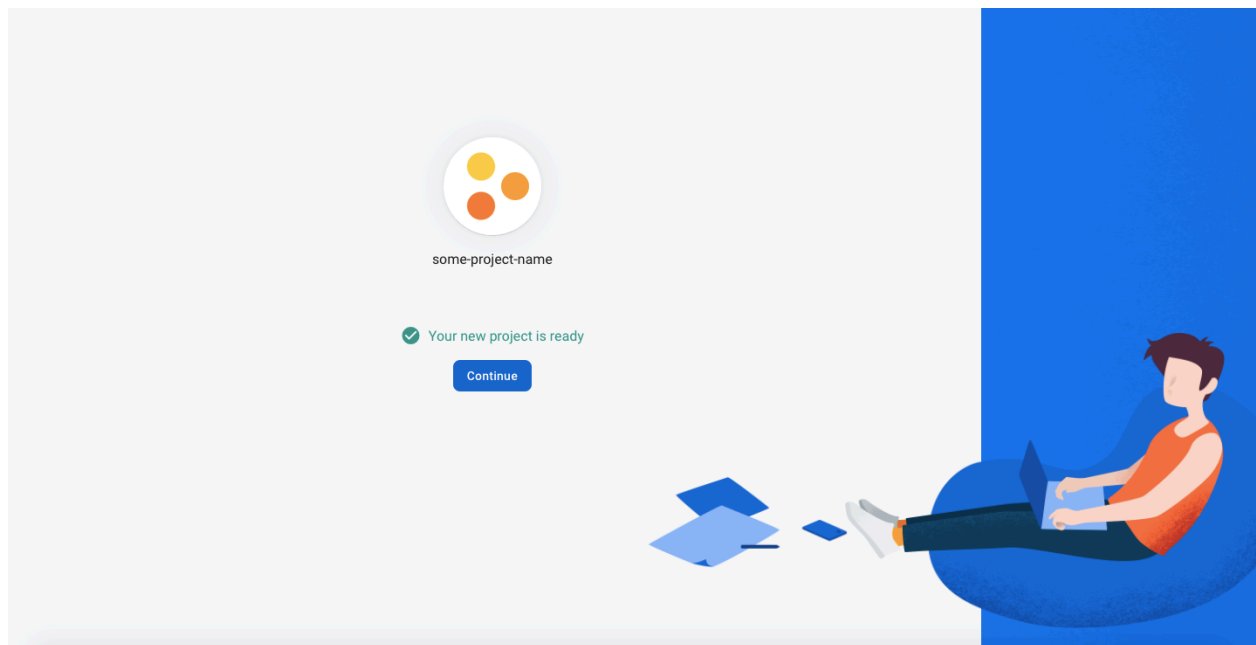


Step 2 (b)

- Next, in dropdown field choose "Default Account for Firebase" and press Continue (Step 3(a)). Then, it will be required to wait a little to let the cloud create the project. In the following step, press Continue (Step 3(b)).

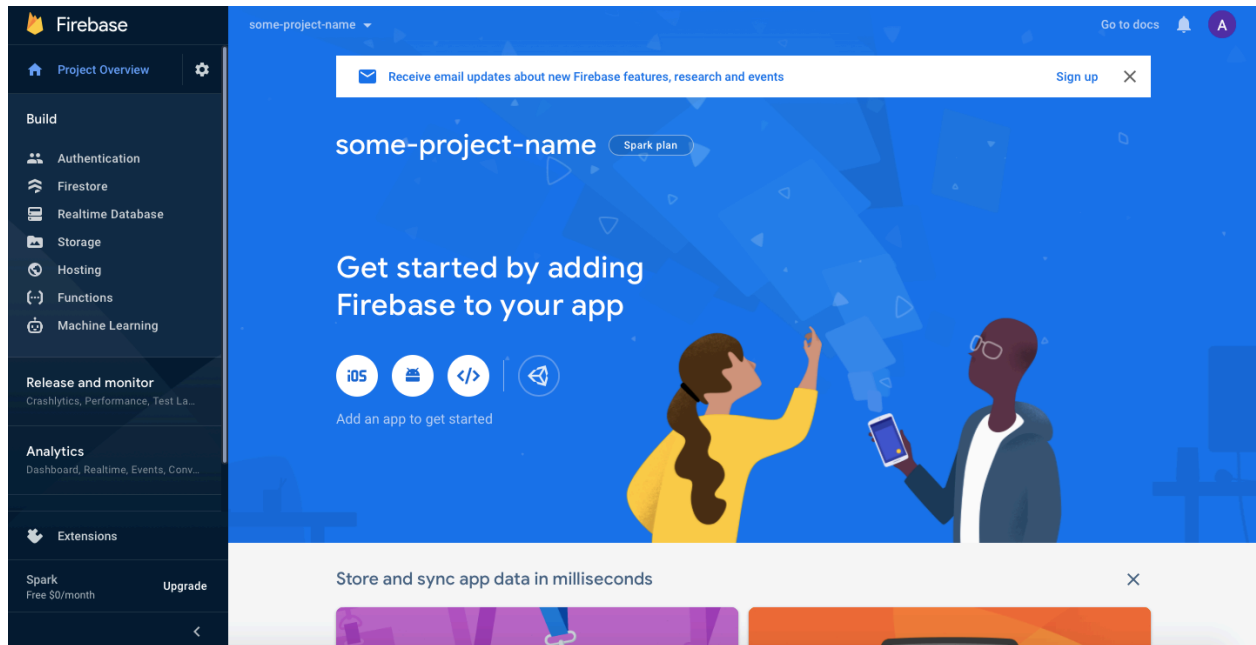


Step 3(a)

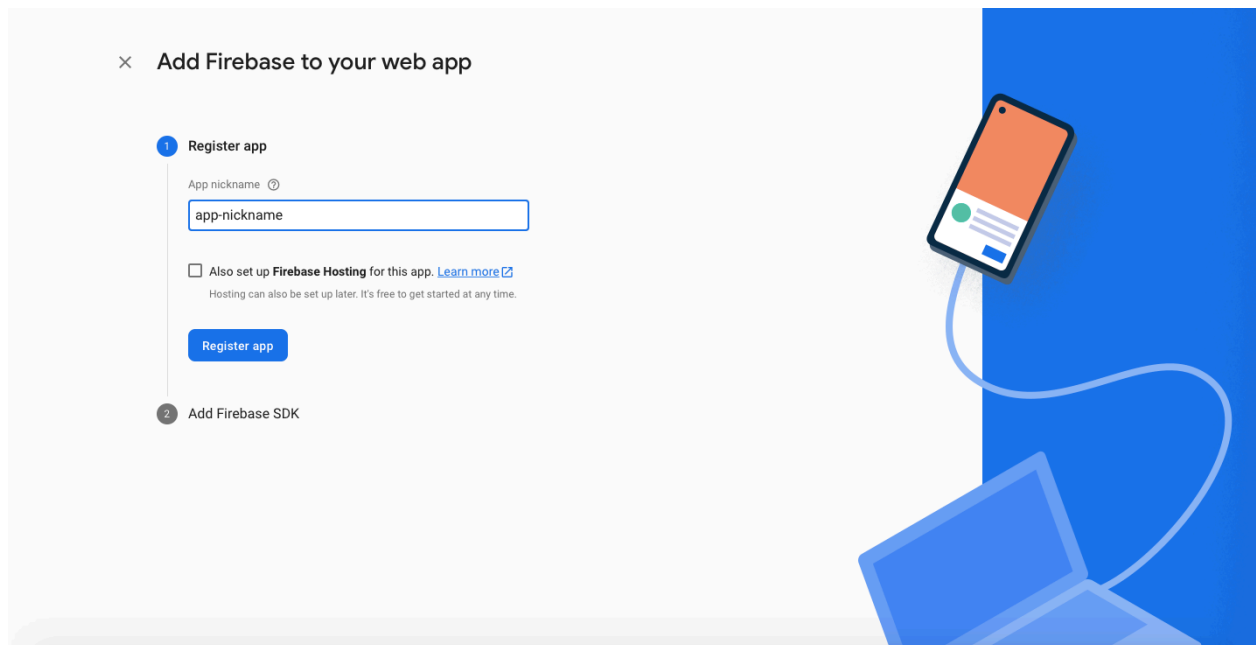


Step 3(b)

4. Once the project is created, you can jump to the configuration of database. In opened page you can see three options (IOS, Android and Web). Click on "Web" icon (Step 4(a)) and you will be forwarded to the page where you should register your app and add nickname to it (Step(b)). Press Register.



Step 4(a)



Step 4(b)

5. Copy the firebaseConfig to your configuration page in code (it will be shown in following pages).

✓ Register app

2 Add Firebase SDK

Copy and paste these scripts into the bottom of your <body> tag, but before you use any Firebase services:

```
<!-- The core Firebase JS SDK is always required and must be listed first -->
<script src="https://www.gstatic.com/firebasejs/8.4.1/firebase-app.js"></script>

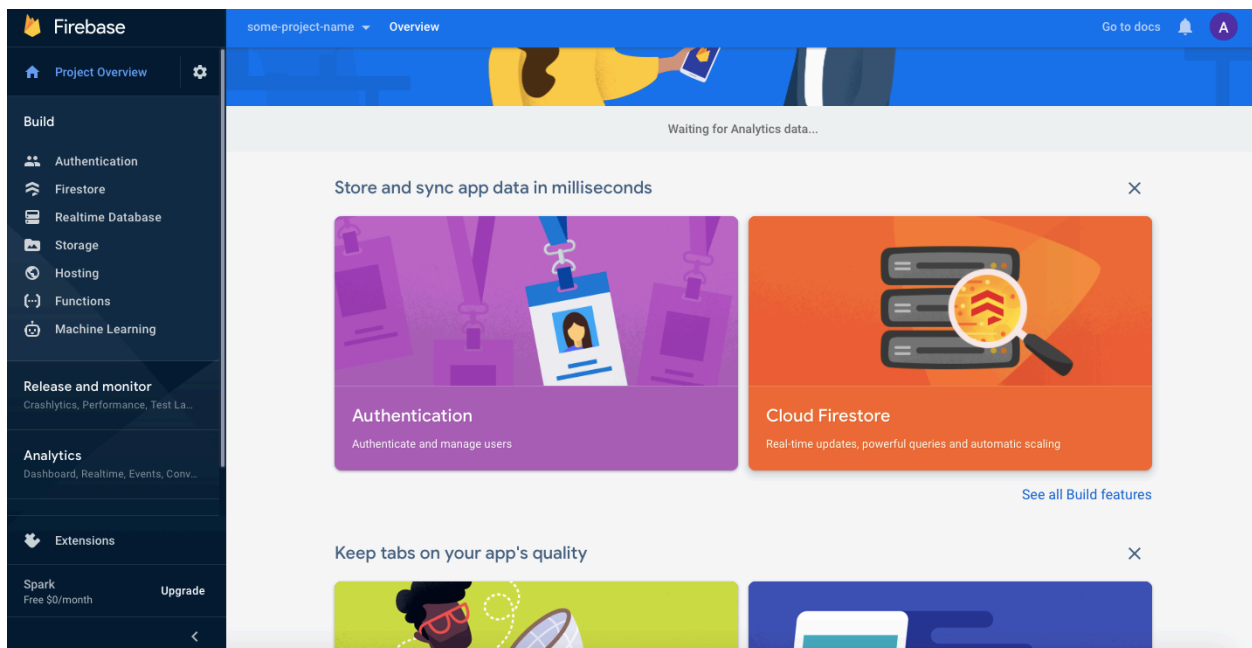
<!-- TODO: Add SDKs for Firebase products that you want to use
https://firebase.google.com/docs/web/setup#available-libraries -->
<script src="https://www.gstatic.com/firebasejs/8.4.1/firebase-analytics.js"></script>

<script>
  // Your web app's Firebase configuration
  // For Firebase JS SDK v7.20.0 and later, measurementId is optional
  var firebaseConfig = {
    apiKey: "AIzaSyAS1fWRsm_6xsr_gsGWrbvNE2b3_mulwI",
    authDomain: "some-project-name-15aa0.firebaseio.com",
    projectId: "some-project-name-15aa0",
    storageBucket: "some-project-name-15aa0.appspot.com",
    messagingSenderId: "282033917388",
    appId: "1:282033917388:web:76ff476bab91c6fdce169",
    measurementId: "G-MV2CDXF8CR"
  };
  // Initialize Firebase
  firebase.initializeApp(firebaseConfig);
  firebase.analytics();
</script>
```

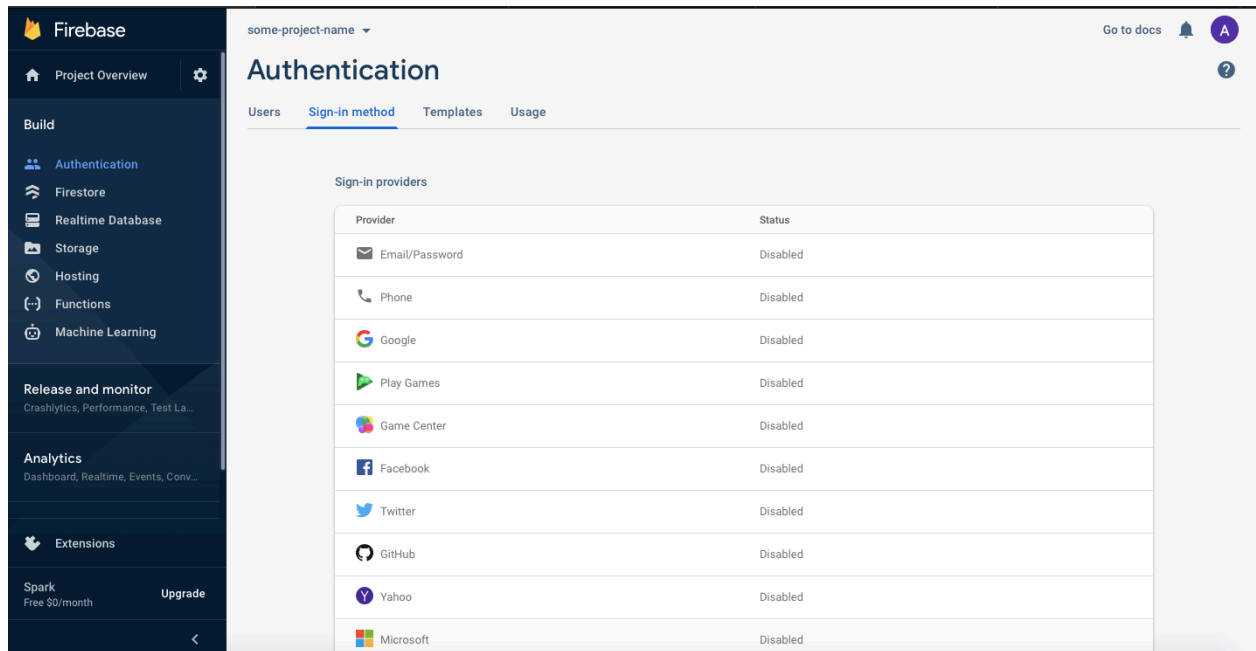
Learn more about Firebase for web: [Get started](#), [Web SDK API reference](#), [Samples](#)



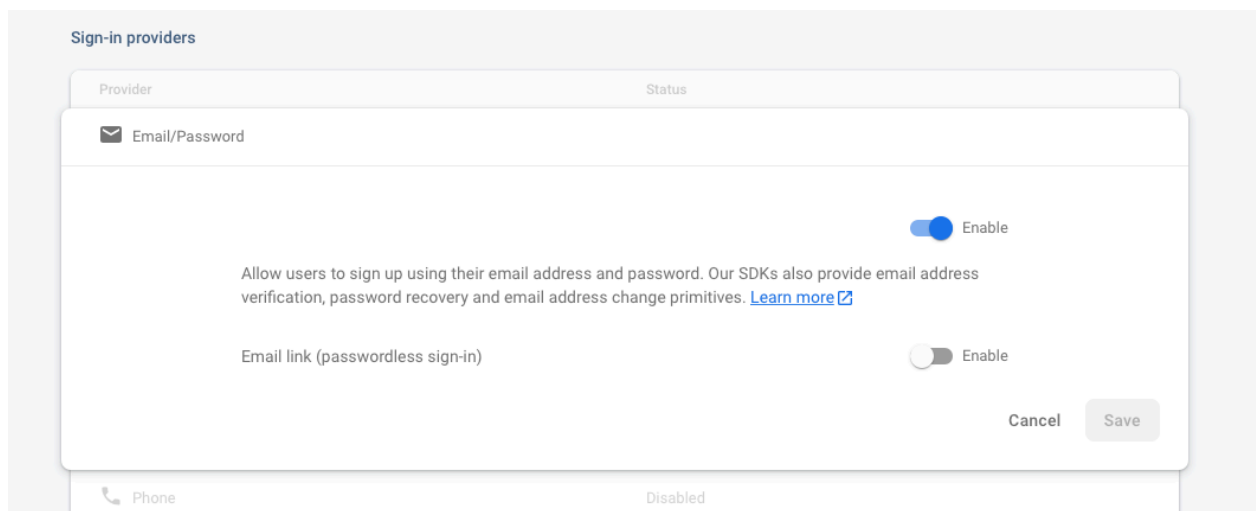
6. Press on continue to console:



7. Choose "Authentication" (Step 6) and in opened page click on "Email/Password" option (Step 7(a)). Then, enable upper icon of two disabled ones (Step 7(b)). Now we have our authentication option of Firebase database working.



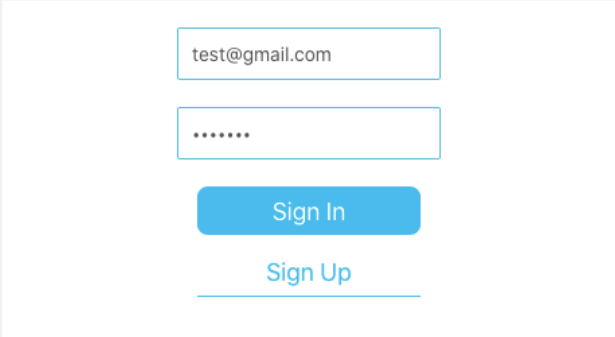
Step 7(a)



Step 7(b)

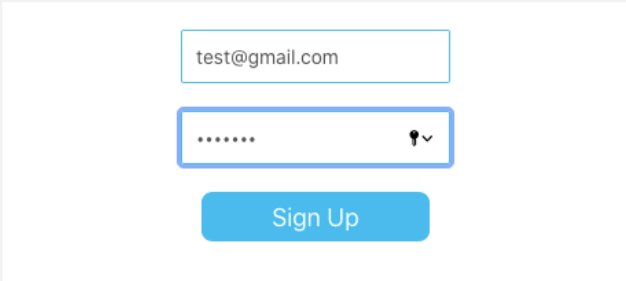
**Create “Sign In” and “Sign Up” options
using Firebase authentication and React Native.**

App



A mockup of a mobile app's Sign In screen. It features a white background with a light gray border. At the top, the word "App" is displayed in the header. The main content area contains a white box with a light gray border. Inside this box, there are two text input fields: the first contains "test@gmail.com" and the second contains "*****". Below the input fields is a blue button with the text "Sign In". Below the button is a blue link with the text "Sign Up" that has a thin blue underline.

← App



A mockup of a mobile app's Sign Up screen. It features a white background with a light gray border. At the top, the word "App" is displayed in the header, preceded by a back arrow icon. The main content area contains a white box with a light gray border. Inside this box, there are two text input fields: the first contains "test@gmail.com" and the second contains "*****". The second input field has a blue border and a small blue icon of a person with a checkmark to its right. Below the input fields is a blue button with the text "Sign Up".

In order to create above app, first it's necessary to install firebase to react app by running "npm install firebase". Once it's installed copy and paste above mentioned firebaseConfig to our App.js file where we configure our database connection and routes between sign in and sign up pages:

```
JS App.js M X
JS App.js > [⌘] navigator > 🔑 Signin
1 | import { createAppContainer } from 'react-navigation';
2 | import { createStackNavigator } from 'react-navigation-stack';
3 | import Signin from './screens/signinScreen';
4 | import Signup from './screens/signupScreen';
5 | import firebase from 'firebase';
6 |
7 |
8 | const firebaseConfig = {
9 |   apiKey: "AIzaSyDvpPjJAXagzJJHu0lPTIGTZZeZqaDw_AY",
10 |   authDomain: "auth-tutorial-4e6c2.firebaseio.com",
11 |   projectId: "auth-tutorial-4e6c2",
12 |   storageBucket: "auth-tutorial-4e6c2.appspot.com",
13 |   messagingSenderId: "920195860730",
14 |   appId: "1:920195860730:web:f1374ea1b09816c8eae143",
15 |   measurementId: "G-GBEM9DQNZC"
16 | }
17 |
18 | if (!firebase.apps.length) {
19 |   firebase.initializeApp(firebaseConfig);
20 | }else {
21 |   firebase.app(); // if already initialized, use that one
22 | }
23 |
```


Sign Up

Using the ***createUserWithEmailAndPassword()*** method, we create a new user and save its credentials:

```

signUp = (email, password) => {
  if (this.state.password.length < 6) {
    alert("Password should be at least 6 characters")
    return;
  }
  firebase.auth().createUserWithEmailAndPassword(email, password);
  alert("Account was created")
}

```

To call the created `signUp` function we assign it to `onPress` in `TouchableOpacity`:

```
<TouchableOpacity  
    |   |   |   |  
    |   |   |   |   onPress = {()=> this.signUp(this.state.email, this.state.password)}  
    |   |   |   |   style = {styles.button}  
    |   |   |   |  
    |   |   |   |>  
    |   |   |   |  
    |   |   |   |   <Text style={styles.buttonText}>Sign Up</Text>  
    |   |   |   |</TouchableOpacity>
```

Sign In

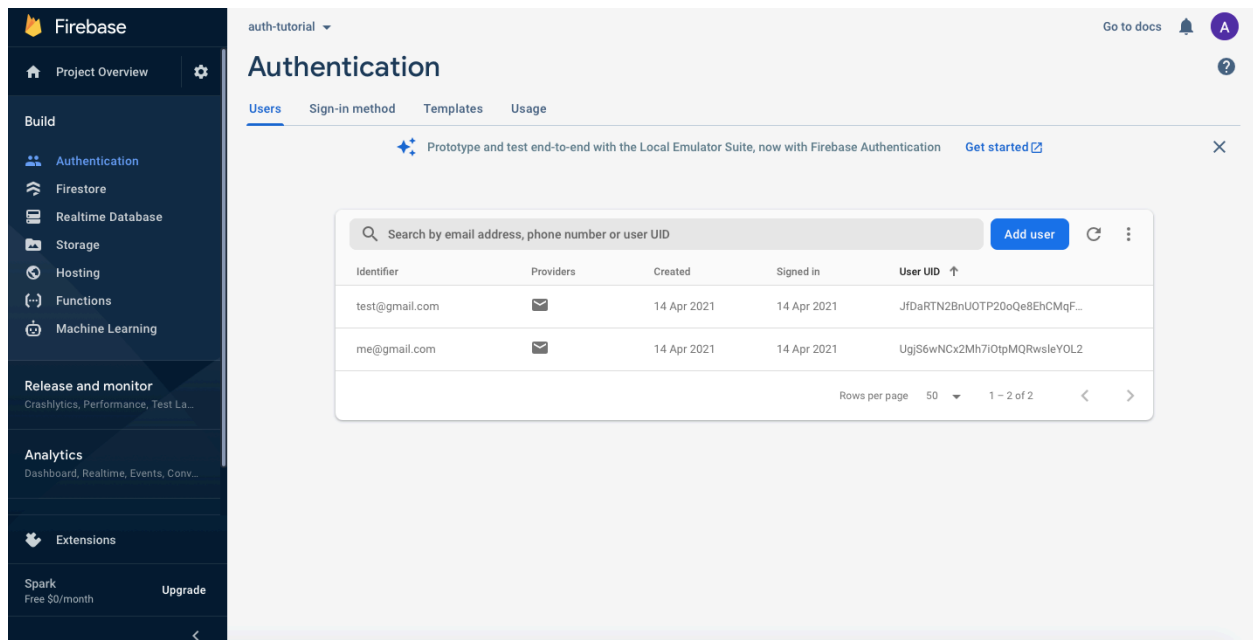
Method ***signInWithEmailAndPassword()*** of the FirebaseAuth instance is used to log in the user:

```
signIn = (email, password)=>{
  firebase.auth().signInWithEmailAndPassword(email, password)
    .then(function(user){
      if(user){
        console.log("logged in");
      }
    })
}
```

To call the created signIn function we assign it to onPress in TouchableOpacity:

```
<TouchableOpacity
  |
  |   onPress = {()=> this.signIn(this.state.email, this.state.password)}
  |   style = {styles.buttonI}
  |
  |   <Text style={styles.buttonText}>Sign In</Text>
  |
  | </TouchableOpacity>
```

To prove that the user was actually created, it's possible to view the registered users in firebase web interface:



The screenshot shows the Firebase Authentication 'Users' page. The left sidebar contains the Firebase logo and navigation links for Project Overview, Build (Authentication, Firestore, Realtime Database, Storage, Hosting, Functions, Machine Learning), Release and monitor, Analytics, Extensions, and Spark. The main content area is titled 'Authentication' and includes tabs for Users, Sign-in method, Templates, and Usage. A search bar at the top allows searching by email address, phone number, or user UID. Below the search bar is a table with two users:

Identifier	Providers	Created	Signed in	User UID ↑
test@gmail.com		14 Apr 2021	14 Apr 2021	JfDaRTN2BnUOTP20oQe8EhCMqF...
me@gmail.com		14 Apr 2021	14 Apr 2021	UgjS6wNCx2Mh7i0tpMQRwsleYOL2

At the bottom of the table, it indicates 'Rows per page 50' and '1 - 2 of 2'.