

# Creating Users

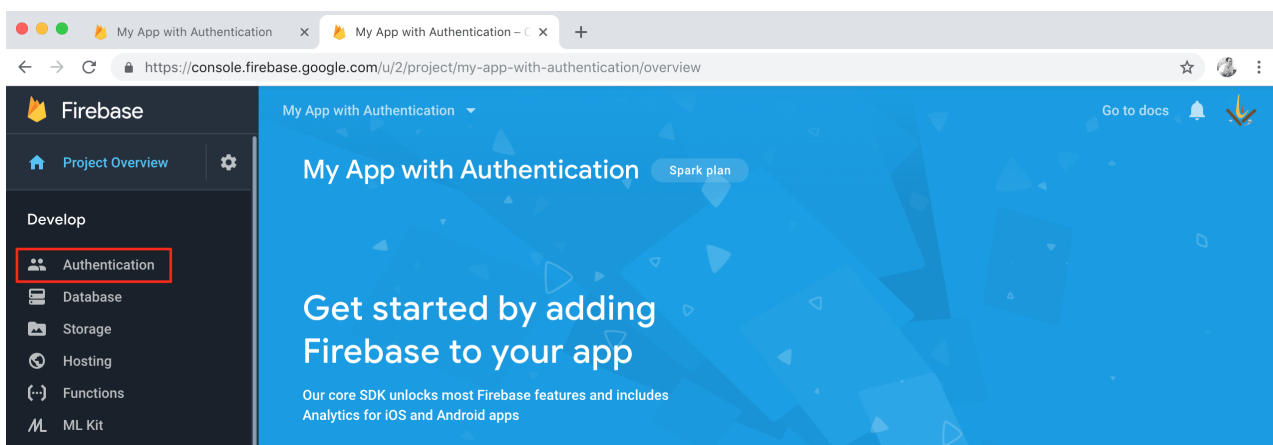
In this lesson, we grab the values from the create user authentication form and pass those values to Firebase in order to create a new user for our app. It will also sign the new user in.

## WE'LL COVER THE FOLLOWING

- Activate email and password sign-in method
  - Sign-in Method
  - Email/Password
- Enable Email/Password
- Create User Form Event, Listener
- `createUserWithEmailAndPassword` firebase method
- `sendPasswordResetEmail()` firebase method
- The Authentication Boilerplate Application

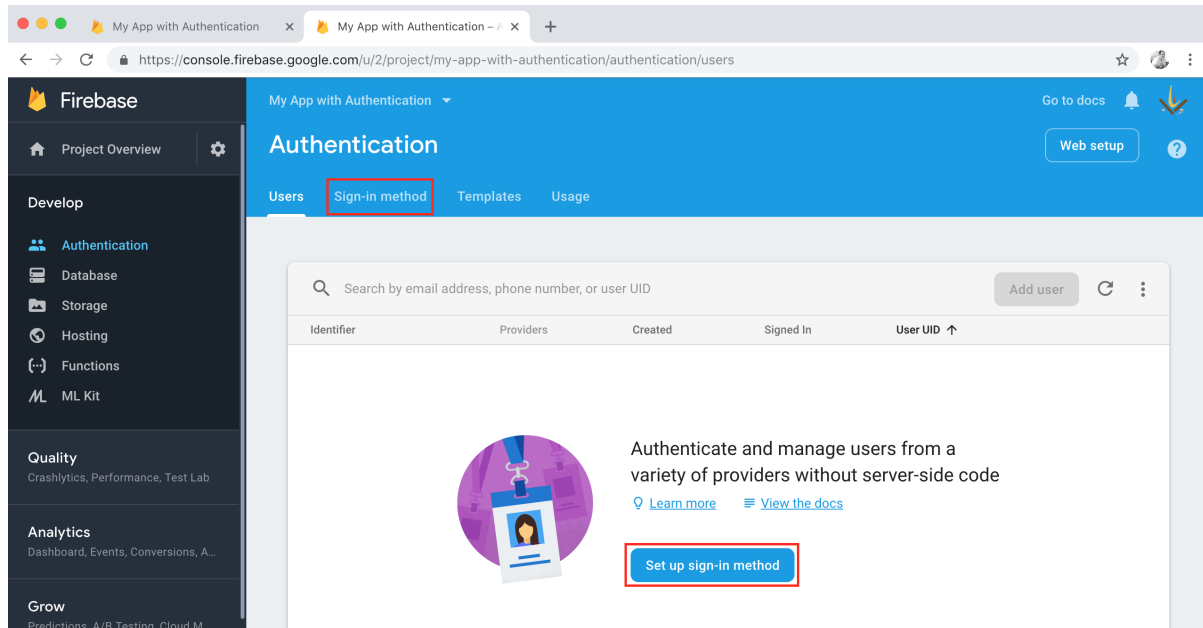
## Activate email and password sign-in method #

Go to the firebase console. Select your app. Once there, you will see the screen given below. On the left, click on **Authentication**.



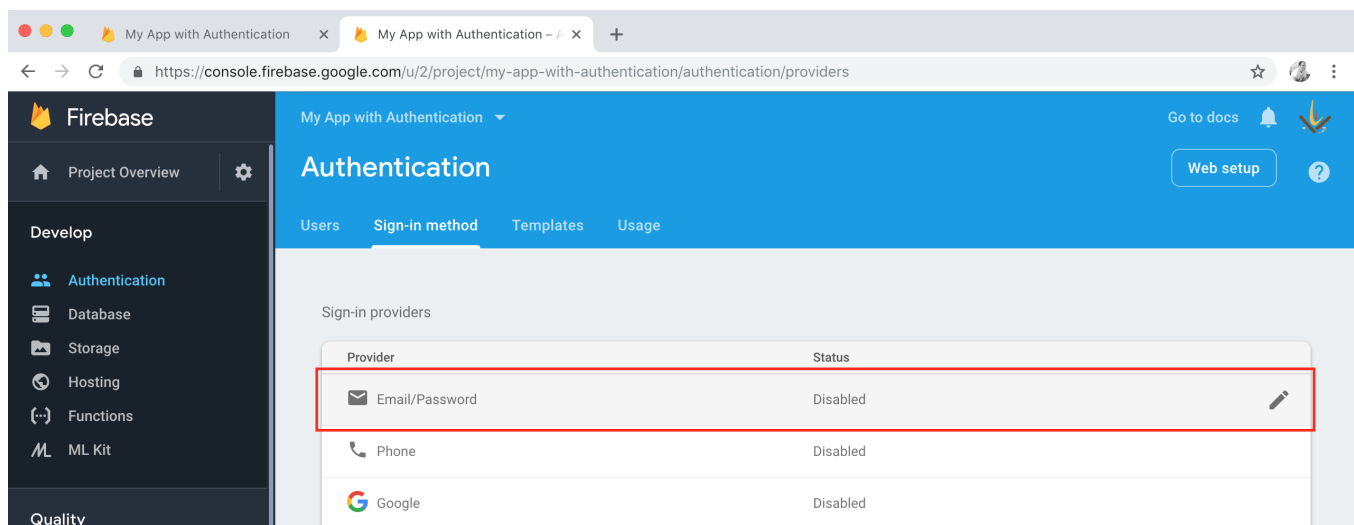
## Sign-in Method #

Click on **Sign-in method** which is to the right of **Users** tab.



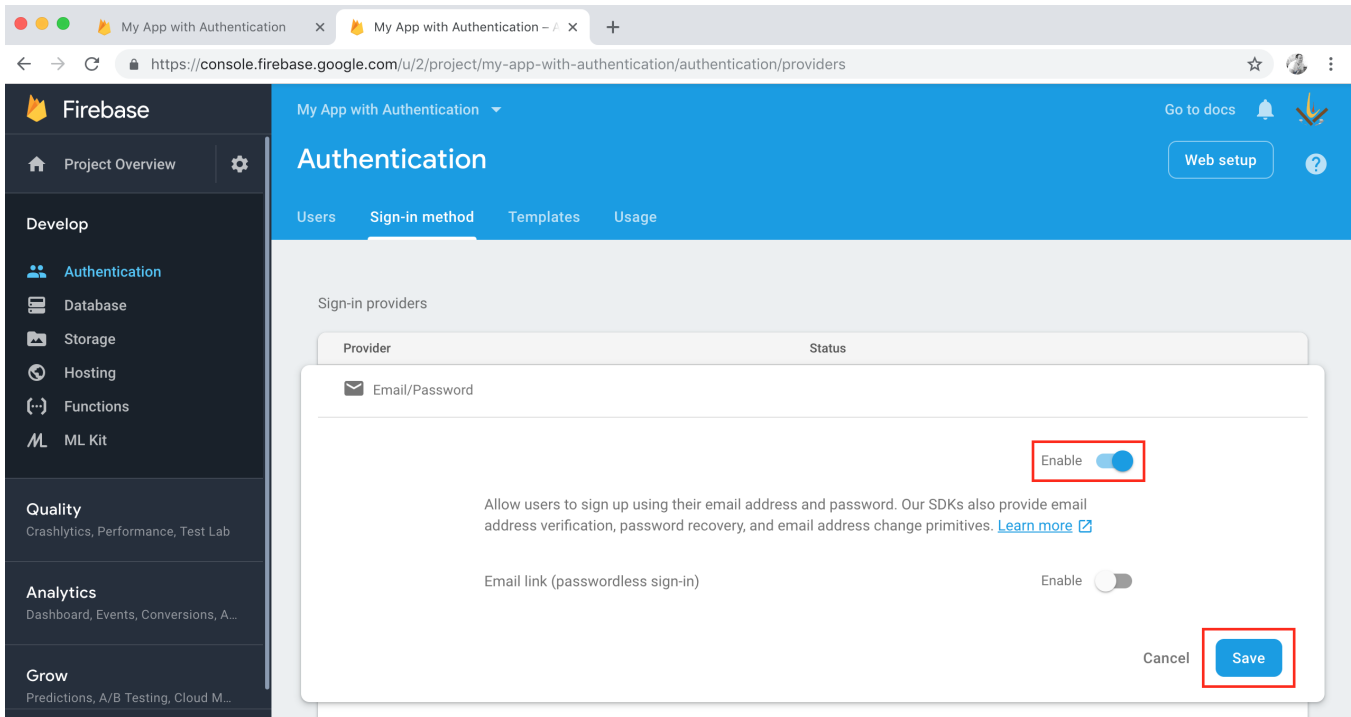
## Email/Password #

Under the Sign-in providers, please select **Email/Password**.



## Enable Email/Password #

Once this menu opens, click on the **Enable** toggle to enable it. Then click on **Save**.



## Create User Form Event, Listener #

All the necessary HTML for the email & password authentication a few lessons ago. Now we need JavaScript to listen to the submit events of each form. Let's start by making the event listener with the create user form. Inside the event listener, we will grab the values from the form.

```
// Create user form submit event
createUserForm.addEventListener(`submit`, event => {
  event.preventDefault();
  // Grab values from form
  const displayName = document.getElementById(`create-user-display-name`).value;
  const email = document.getElementById(`create-user-email`).value;
  const password = document.getElementById(`create-user-password`).value;
});
```

JavaScript

## createUserWithEmailAndPassword firebase method #

We invoke the `createUserWithEmailAndPassword()` Firebase method inside the newly created event listener. On line 9, we are sending the email and password to Firebase which creates a user. On lines 11 - 13, after the new user is created, we update their `displayName`. The user has to exist before assigning a `displayName`. This is why we make another request to Firebase to assign a

display name to the current user.

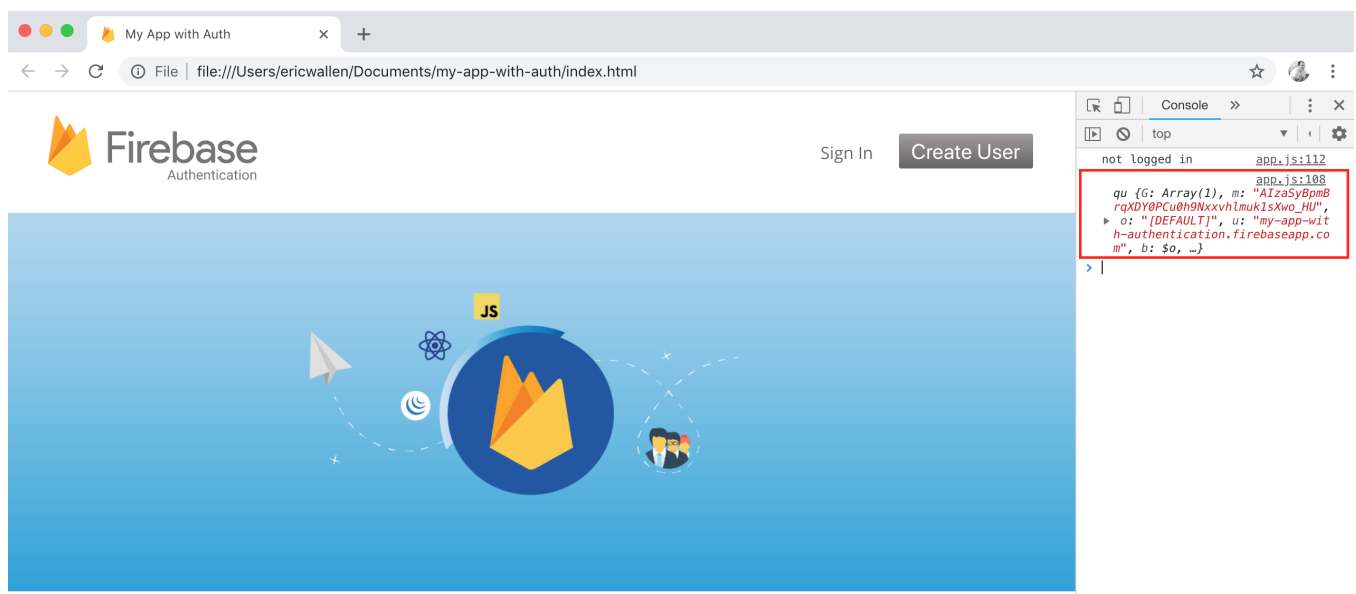
After the user is created clear the inputs of their values with the built-in JavaScript method, `reset`. We do this because when you sign out, then sign-in again you would see the values from the original sign-in and your details would still be in the inputs. Not clearing the inputs is a security concern.

```
// Create user form submit event
createUserForm.addEventListener(`submit`, event => {
  event.preventDefault();
  // Grab values from form
  const displayName = document.getElementById(`create-user-display-name`).value;
  const email = document.getElementById(`create-user-email`).value;
  const password = document.getElementById(`create-user-password`).value;
  // Send values to Firebase
  auth.createUserWithEmailAndPassword(email, password)
    .then(() => {
      auth.currentUser.updateProfile({
        displayName: displayName
      })
      createUserForm.reset();
      hideAuthElements();
    })
    .catch(error => {
      console.log(error.message);
    });
});
```

JavaScript

## `sendPasswordResetEmail()` firebase method #

We invoke `sendPasswordResetEmail()` firebase method inside the newly created event listener.



# The Authentication Boilerplate Application #

Go ahead and create a new user with the authentication form. There will be a visual indication in the header of your app that the user was created and simultaneously signed in because the **sign in** and **create user** buttons will disappear. Also, as the developer, you can check your console. You will either see the `user object` or *not signed in* depending on the authentication state.

This code requires the following keys to execute: ^

Key:	Value:
apiKey	Not Specified...
authDomain	Not Specified...
databaseURL	Not Specified...
projectId	Not Specified...
storageBucket	Not Specified...
messagingSenderId	Not Specified...
appId	Not Specified...

Output

JavaScript

HTML

CSS (SCSS)



# Authentication Boilerplate

Sign In

Create User



Console

Clear

After you see your user as logged in using the create user form, you have completed this lesson.

In the next lesson, you will learn how to show an authentication-based user dashboard for your application to users.