

Signing In

In this lesson, we grab the values from the sign in authentication form and pass those values to Firebase to authenticate already existing users of our app.

WE'LL COVER THE FOLLOWING



- Sign In Form Event Listener
- `signInWithEmailAndPassword` Firebase Method
- The Authentication Boilerplate Application

Sign In Form Event Listener

```
// Sign in form submit event
signInForm.addEventListener(`submit`, event => {
  event.preventDefault();
  // Grab values from form
  const email = document.getElementById(`sign-in-email`).value;
  const password = document.getElementById(`sign-in-password`).value;
});
```



JavaScript

`signInWithEmailAndPassword` Firebase Method

We invoke the `signInWithEmailAndPassword()` Firebase method inside the newly created event listener.

```
// Sign in form submit event
signInForm.addEventListener('submit', event => {
  event.preventDefault()
  // Grab values from form
  const email = document.getElementById('sign-in-email').value
  const password = document.getElementById('sign-in-password').value
  // Send values to Firebase
  auth.signInWithEmailAndPassword(email, password)
  .then(() => {
    signInForm.reset()
    hideAuthElements()
  })
});
```



```
.catch(error =>{  
  console.log(error.message)  
  
  })  
})
```

JavaScript

The Authentication Boilerplate Application

Now use the **sign-in form** to sign back in with a previously created user.

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...
appld	Not Specified...

Output

JavaScript

HTML

CSS (SCSS)



Console

Clear

In the next lesson, we give our users the ability to reset their passwords

In the next lesson, we give our users the ability to reset their passwords.