

module-10_1

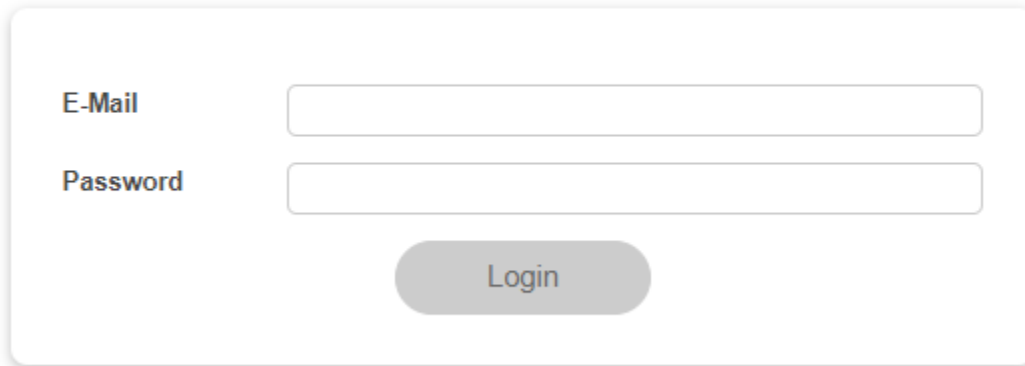
Project Name: Email & Password validation using useEffect.

[module-10_1: module-10_1 is about useEffect.

useEffect only re-renders when its dependency changes.

an important react hook,

A Typical Page

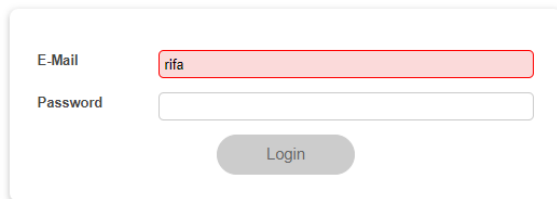


A typical login page form. It features two input fields: "E-Mail" and "Password". Below these fields is a "Login" button. The form is enclosed in a light gray rounded rectangle.

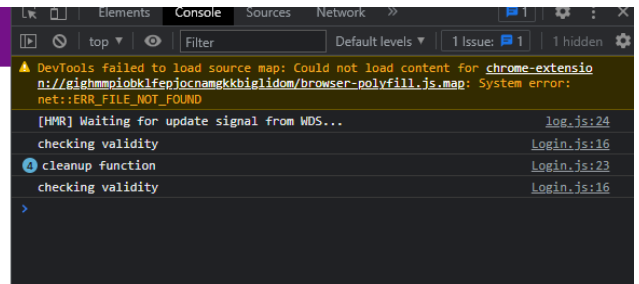
useEffect returns a function that runs only before the re-execution of useEffect hook,

** check the last picture for better understanding,

A Typical Page



A typical login page form, similar to the one above, but with a red border around the "E-Mail" input field, indicating a validation error. The "Password" field and "Login" button are also present.



```
JS App.js    JS Login.js    X    ≡ module-10.txt
module-10_1 > src > components > Login > JS Login.js > [x] Login
7  const Login = (props) => {
8      const [enteredEmail, setEnteredEmail] = useState('');
9      const [emailIsValid, setEmailIsValid] = useState();
10     const [enteredPassword, setEnteredPassword] = useState('');
11     const [passwordIsValid, setPasswordIsValid] = useState();
12     const [formIsValid, setFormIsValid] = useState(false);
13
14     useEffect(() => {
15         const identifier = setTimeout(() => {
16             console.log('checking validity');
17             setFormIsValid(
18                 enteredEmail.includes('@') && enteredPassword.trim().length > 6
19             );
20         }, 2000);
21
22         return () => {
23             console.log('cleanup function');
24             clearTimeout(identifier);
25         };
26     }, [enteredEmail, enteredPassword]);
27 }
```

Let's, check the code & the console in the above picture, checking validity runs only once, because, when we give input in the form, dependency of useEffect changes & re-renders. But before re-render, it goes to return function of useEffect & previous setTimeout is destroyed by clearTimeout. And then when useEffect re-renders, new timeout is set again and so on. This is really interesting,