

Guía de pruebas Aplicación Superhéroes

Documentación oficial de jest que es una librería para hacer pruebas en react <https://jestjs.io/>

Documentación oficial enzyme <https://enzymejs.github.io/enzyme/>

Documentación oficial enzyme to json <https://www.npmjs.com/package/enzyme-to-json>

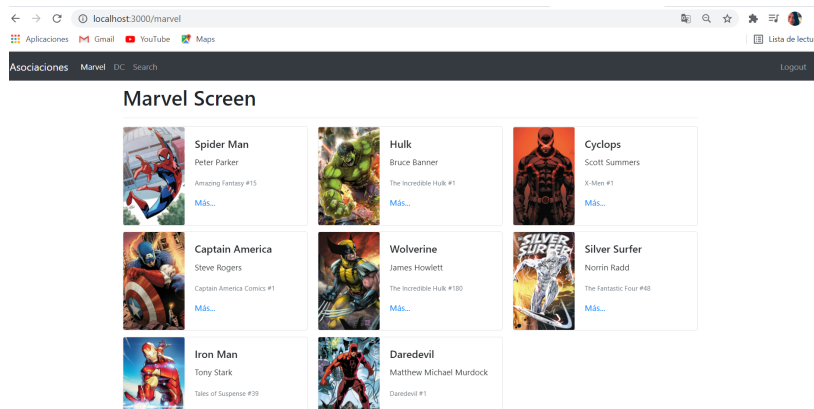
Descargamos la aplicación de heroes

<https://github.com/jennymontoya1001/react-hero-app-testing.git>

La abrimos en visual studio code

Corremos la instrucción npm install para instalar los módulos de node

Corremos la instrucción npm start



Abrimos otra terminal

Instalamos enzyme

Docuemntación enzyme <https://enzymejs.github.io/enzyme/>

```
npm i --save-dev enzyme enzyme-adapter-react-16
```

Instalamos enzume to JSON

Documentación enzyme to json <https://www.npmjs.com/package/enzyme-to-json>

```
npm install --save-dev enzyme-to-json
```

Dentro de la carpeta src creamos el archivo que se llame setupTests.js dentro de este archivo debemos realizar la configuración de enzyme y enzyme to JSON

De la documentación de enzyme

```
import Enzyme from 'enzyme';
import Adapter from 'enzyme-adapter-react-16';

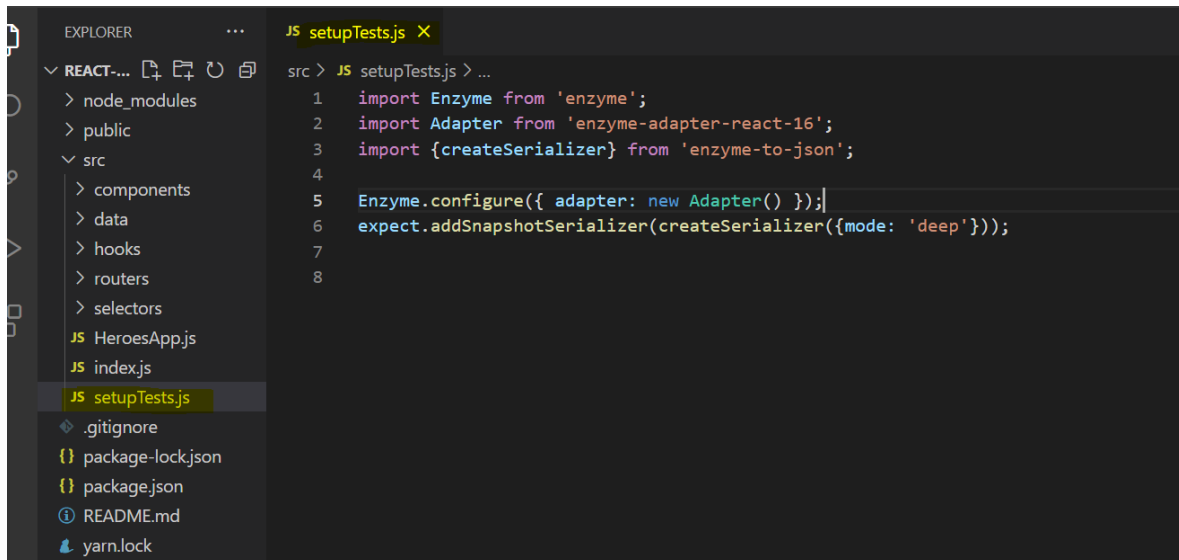
Enzyme.configure({ adapter: new Adapter() });
```

De la documentación de enzyme JSON

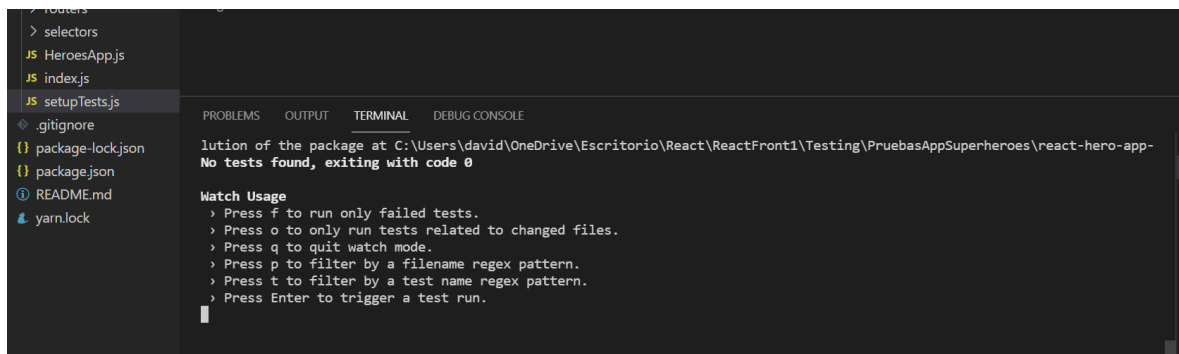
```
import {createSerializer} from 'enzyme-to-json';

expect.addSnapshotSerializer(createSerializer({mode:
'deep'}));
```

El archivo nos debe quedar de la siguiente manera



En nuestra terminal corremos la instrucción npm run test para asegurarnos que nuestro test suite esté pasando

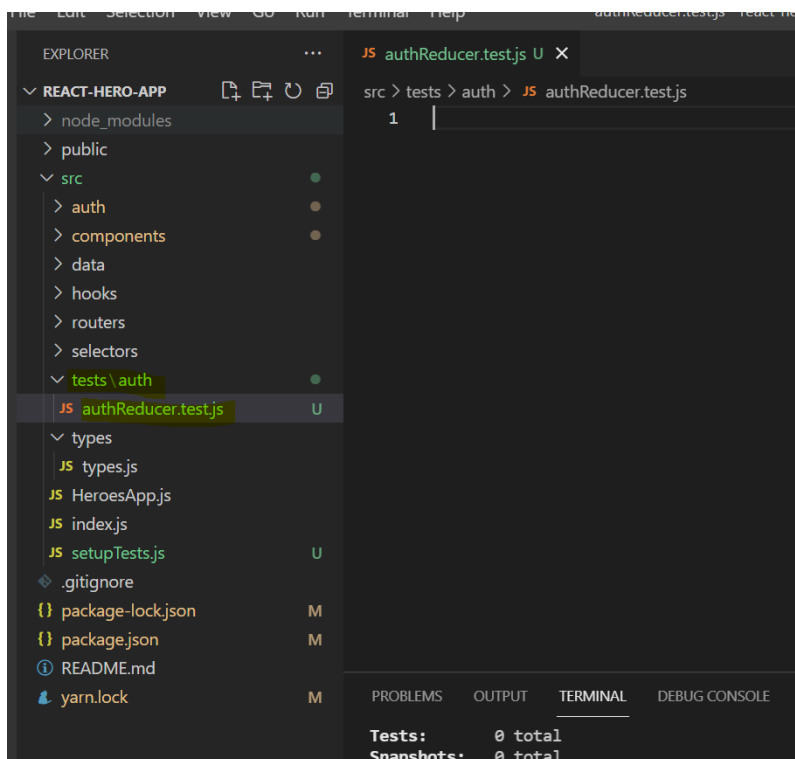


Iniciaremos nuestras pruebas

En nuestro reducer authReducer

Dentro de la carpeta src creamos una carpeta llamada tests, dentro de esta carpeta crearemos la misma estructura que tenemos en nuestro proyecto

Dentro de la carpeta test creamos una carpeta llamada auth y dentro de ella un archivo llamado authReducer.test.js



Dentro de este archivo vamos a realizar las siguientes pruebas:

Nombre prueba 1: debe de retornar el estado por defecto

Nombre prueba 2: debe de autenticar y colocar el nombre del usuario

Nombre prueba 3: debe borrar el nombre del usuario y la propiedad logged en false

Vamos con nuestra primera prueba

El reducer siempre regresa un estado

El estado inicial es false

Como acción le enviamos un objeto vacío, es decir que utilice el valor por defecto. Si el estado inicial es falso, se espera que sea falso

```
JS authReducer.test.js U X
src > tests > auth > JS authReducer.test.js > describe('Pruebas en authReducer') callback > test('debe de retornar el estado p
1 import authReducer from '../auth/authReducer';
2
3 describe('Pruebas en authReducer', () =>{
4
5     test('debe de retornar el estado por defecto', () => {
6
7         // el reducer siempre regresa un estado
8         //Estado inicial en false
9         const state = authReducer({ logged: false}, {});
10        //Yo espero que el estado sea falso si el estado inicial es falso
11        expect( state ).toEqual({logged: false});
12    })
13 })
```

Nuestra prueba no pasa, ya que nuestro authReducer no tiene el retorno por defecto

```
5 test('debe de retornar el estado por defecto', () => {
6
7     // el reducer siempre regresa un estado
8     //Estado inicial en false
9     const state = authReducer({ logged: false}, {});
10    //Yo espero que el estado sea falso si el estado inicial es falso
11    expect( state ).toEqual({logged: false});
12 })

at Object.<anonymous> (src/tests/auth/authReducer.test.js:9:23)
at processTicksAndRejections (node:internal/process/task_queues:96:5)

Test Suites: 1 failed, 1 total
Tests: 1 failed, 1 total
Snapshots: 0 total
Time: 6.666s
Ran all test suites related to changed files.

Watch Usage: Press w to show more.
```

Nos vamos para el archivo authReducer y le cambiamos el retorno por defecto

```
JS authReducer.test.js U JS authReducer.js M X
src > auth > JS authReducer.js > [0] default
1 import { types } from '../types/types';
2
3
4
5 const authReducer = (state = {}, action) => {
6
7     switch ( action.type ) {
8         case types.login:
9             return {
10                 ...action.payload,
11                 logged: true
12             }
13
14         case types.logout:
15             return {
16                 logged: false
17             }
18
19         default:
20             return state;
21     }
22
23 }
24
25 export default authReducer;
```

Nuestra prueba pasa

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Snapshots: 0 total
Time: 7.505s
Ran all test suites related to changed files.
PASS src/tests/auth/authReducer.test.js
Pruebas en authReducer
✓ debe de retornar el estado por defecto (4ms)
Test Suites: 1 passed, 1 total
Tests: 1 passed, 1 total
Snapshots: 0 total
Time: 3.574s
Ran all test suites related to changed files.
Watch Usage: Press w to show more.
```

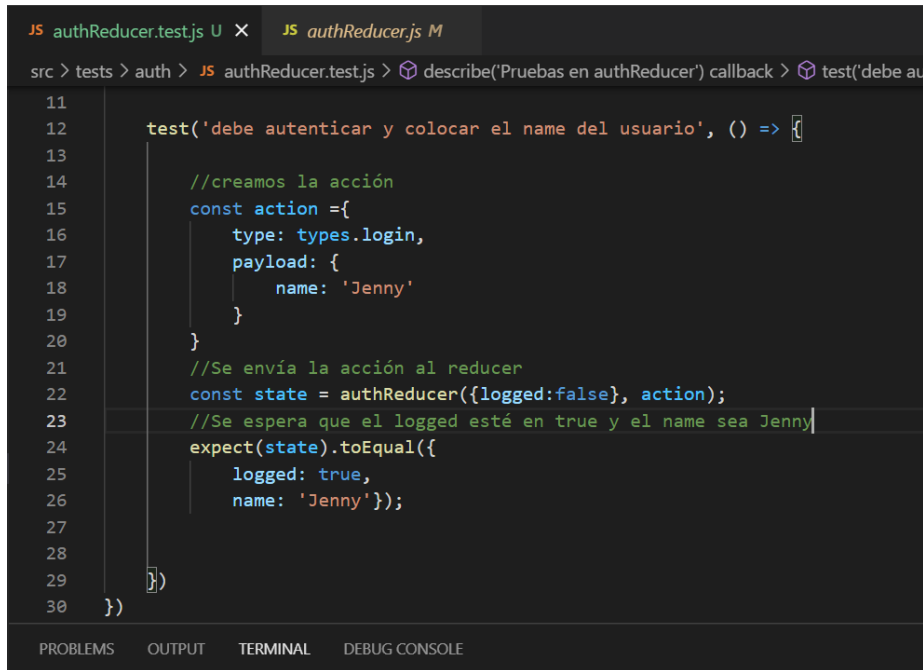
Vamos con nuestra segunda prueba

En este si necesitamos enviar una acción, que debe autenticar y poner el nombre del usuario

Enviamos la acción al reducer , no importa cual sea el estado inicial, por que siempre vamos a enviar un nuevo estado

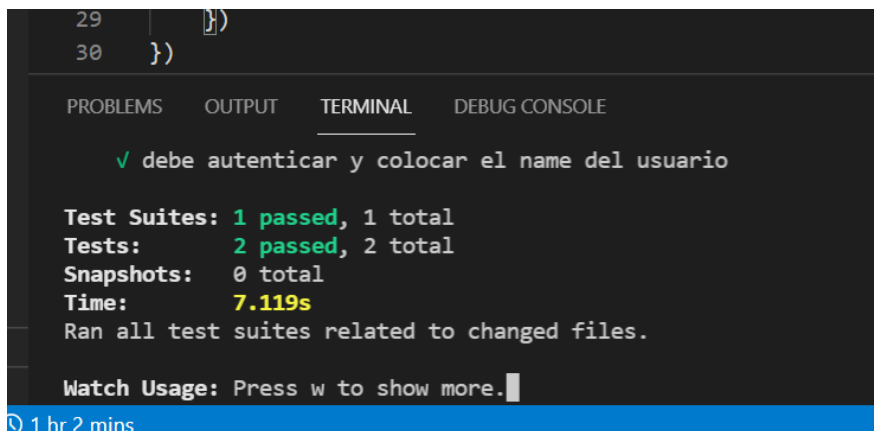
Lo que se esperaría es que el logged esté en true y el name sea Jenny

No nos olvidemos de importar types del archivo types.js



```
11
12   test('debe autenticar y colocar el name del usuario', () => {
13
14     //creamos la acción
15     const action = {
16       type: types.login,
17       payload: {
18         name: 'Jenny'
19       }
20     }
21     //Se envía la acción al reducer
22     const state = authReducer({logged:false}, action);
23     //Se espera que el logged esté en true y el name sea Jenny
24     expect(state).toEqual({
25       logged: true,
26       name: 'Jenny'});
27
28   })
29
30 })
```

Nuestra prueba pasa



```
29   })
30 })

PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

  ✓ debe autenticar y colocar el name del usuario

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:  0 total
Time:        7.119s
Ran all test suites related to changed files.

Watch Usage: Press w to show more.
```

Ahora vamos con nuestra tercera prueba

```
... JS authReducer.test.js U X JS authReducer.js M
src > tests > auth > JS authReducer.test.js > describe('Pruebas en authReducer') callback > test('debe de borrar el nombre de
30
31 test('debe de borrar el nombre del usuario y logged en false', () => {
32
33
34     const action = {
35       type: types.logout,
36     }
37
38     //Estado por defecto
39     const state = authReducer({logged:false, name: 'Jenny'}, action);
40     //Se espera que luego de ejecutar esa acción, el state
41     //sea el logged en false y que no tenga el usuario
42     expect(state).toEqual({
43       logged: false});
44
45
46   })
47 })
```

Nuestra tercera prueba pasa

```
U
M
M
M
U
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
✓ debe de borrar el nombre del usuario y logged en false

Test Suites: 1 passed, 1 total
Tests: 3 passed, 3 total
Snapshots: 0 total
Time: 8.102s
Ran all test suites.

Watch Usage: Press w to show more.
1 hr 18 mins
```