Testing Redux code is easy as we mostly write functions, and most of them are pure. So we can test it without even mocking them. Here, we are using JEST as a testing engine. It works in the node environment and does not access DOM.

We can install JEST with the code given below −

npm install --save-dev jest

With babel, you need to install **babel-jest** as follows −

npm install --save-dev babel-jest

And configure it to use babel-preset-env features in the .babelrc file as follows −

{

"presets": ["@babel/preset-env"]

}

And add the following script in your package.json:

{

//Some other code

"scripts": {

//code

"test": "jest",

"test:watch": "npm test -- --watch"

},

//code

}

Finally, **run npm test or npm run test**. Let us check how we can write test cases for action creators and reducers.

Test Cases for Action Creators

Let us assume you have action creator as shown below −

export function itemsRequestSuccess(bool) {

return {

type: ITEMS\_REQUEST\_SUCCESS,

isLoading: bool,

}

}

This action creator can be tested as given below −

import \* as action from '../actions/actions';

import \* as types from '../../constants/ActionTypes';

describe('actions', () => {

it('should create an action to check if item is loading', () => {

const isLoading = true,

const expectedAction = {

type: types.ITEMS\_REQUEST\_SUCCESS, isLoading

}

expect(actions.itemsRequestSuccess(isLoading)).toEqual(expectedAction)

})

})

Test Cases for Reducers

We have learnt that reducer should return a new state when action is applied. So reducer is tested on this behaviour.

Consider a reducer as given below −

const initialState = {

isLoading: false

};

const reducer = (state = initialState, action) => {

switch (action.type) {

case 'ITEMS\_REQUEST':

return Object.assign({}, state, {

isLoading: action.payload.isLoading

})

default:

return state;

}

}

export default reducer;

To test above reducer, we need to pass state and action to the reducer, and return a new state as shown below −

import reducer from '../../reducer/reducer'

import \* as types from '../../constants/ActionTypes'

describe('reducer initial state', () => {

it('should return the initial state', () => {

expect(reducer(undefined, {})).toEqual([

{

isLoading: false,

}

])

})

it('should handle ITEMS\_REQUEST', () => {

expect(

reducer(

{

isLoading: false,

},

{

type: types.ITEMS\_REQUEST,

payload: { isLoading: true }

}

)

).toEqual({

isLoading: true

})

})

})