Home.js

import React from 'react'

export default function Home() {

  return (

    <div className='hmlclass'>Home</div>

  )

}

Home.test.js

import  {shallow}  from "enzyme";

import Home from './Home';

describe("testing home com ", ()=>

{

  it('testcase1forhome', ()=>

  {

    let wrapper =  shallow(<Home/>)

    console.log(wrapper.debug())

    expect(wrapper.exists('.hmlclass')).toEqual(true)

  })

})

AboutUs.js

import React, { Component } from 'react'

export default class About extends Component {

  constructor(props)

  {

    super(props)

    this.state={

      name:'ajay',

      age:20

    }

  }

  render() {

    return (

      <div>About</div>

    )

  }

}

Test.js

import {shallow} from "enzyme"

import About from './About';

describe('test stuite for about', () => {

it('testcase1', ()=>

{

  let wrapper=   shallow(<About/>)

  expect(wrapper.state()).toEqual({name:'ajay',age:20})

})

it('testcase2', ()=>

{

  let wrapper=   shallow(<About/>)

  console.log("test case 2")

  expect(wrapper.contains(<div>About</div>)).toEqual(true)

})

it('testcase3', ()=>

{

  let wrapper=   shallow(<About/>)

  expect(wrapper.containsMatchingElement(<div>About</div>)).toEqual(true)

})

it('testcase4', ()=>

{

  let wrapper=   shallow(<About/>)

  expect(wrapper.containsAnyMatchingElements([(<div>About</div>),(<div>About1</div>)])).toEqual(true)

})

})

Blog.js

import React from 'react'

export default function Blog() {

  return (

    <div className='mainClass'>

          <ul>

            <li key="key1" className="liClass"> Angular </li>

            <li key="key2" className="liClass"> React </li>

            <li key="key3" className="liClass"> Express </li>

          </ul>

         <div className='numbers'>

          <span> one </span>

          <span> two </span>

         </div>

         <div>

          <button className= "my-button disabled"> Submit </button>

s         </div>

    </div>

  )

}

Test.js

import {shallow} from "enzyme"

import Blog from './Blog';

describe.skip('first', () => {

  it('testcase1' , ()=>

  {

  let wrapper = shallow(<Blog/>)

  console.log(wrapper.debug())

   expect(wrapper.find('.liClass').length).toBe(3)

  })

  it('testcase2' , ()=>

  {

  let wrapper = shallow(<Blog/>)

  expect(wrapper.find('.liClass').at(0).key()).toBe("key1")

  })

  // testing span element

  it('testcase3' , ()=>

  {

  let wrapper = shallow(<Blog/>)

  expect(wrapper.find('.numbers').childAt(0).type()).toBe('span');

  })

  it('testcase4' , ()=>

  {

  let wrapper = shallow(<Blog/>)

  expect(wrapper.find('ul').children().length).toBe(3)

  })

  it('testcase5' , ()=>

  {

  let wrapper = shallow(<Blog/>)

  expect(wrapper.find('.my-button').hasClass('disabled')).toBe(true)

  })

 })

ApiCall.js

import React, { Component } from 'react'

import axios from "axios"

export default class ApiCall extends Component {

  state={

     title:"",

     body:""

  }

   componentDidMount()

  {

       this.func(2,3)

  }

  func = async() =>

  {

    let url ="";

    let options=

    {

       method:'GET',

       url:"https://jsonplaceholder.typicode.com/posts"

    }

    try {

       let response = await axios(options)

// console.log(response.json());

     this.setState(

      {

        // age:response.data.studentAge,

        // name:response.data.name,

        // error:false

        title:response.title,

        body: response.body

      }

     )

      console.log(response)

    } catch (error){

      console.log(error)

    } }

  render() {

    return (

      <div>ApiCall

      </div>

    )

  }

}

Test.js

import {shallow} from "enzyme"

import ApiCall from "./ApiCall"

import mockAxios from "axios"

import \* as math from './Mat';

jest.mock('axios')

describe("testing api ", ()=>

{

  it("testcase1", ()=>

  {

  let wrapper = shallow(<ApiCall/>)

  console.log(wrapper.state())

  jest.spyOn(mockAxios, 'default').mockResolvedValue(

    {

        title:'title1',

        body:"body1"

    }

  )

  console.log(wrapper.state())

  // console.log({data})

  return wrapper.instance().func().then(data=>

    {

      console.log(wrapper.state())

      expect(wrapper.state()).toEqual({title:'title1', body:'body1'})

    })

})

it('math function' , ()=>

{

  expect(math.product(1,2)).toEqual(2)

  math.add = jest.fn().mockImplementation(add =>25);

  math.subtract = jest.fn().mockImplementation(subtract =>10);

  expect(math.add(12,12)).toEqual(25)

  expect(math.subtract(10,12)).toEqual(10)

})

})