npm i eventsourcemock

*// \_\_test\_\_/configJSDom.js*

**import** EventSource **from** 'eventsourcemock';

Object.defineProperty(window, 'EventSource', {

  value**:** EventSource,

});

// package.json

"jest": {

  "setupFiles": [

    "./\_\_test\_\_/configJSDom.js"

  ]

}

//test file

import React from 'react';

import { cleanup, render, screen, act } from '@testing-library/react';

import userEvent from '@testing-library/user-event';

import { mockIonicReact, waitForIonicReact } from '@ionic/react-test-utils';

import { ionFireEvent as fireEvent } from '@ionic/react-test-utils';

import "jest-canvas-mock";

import 'jest-webgl-canvas-mock';

import mockAxios from "axios";

import { [ReplaySubject](https://rxjs.dev/api/index/class/ReplaySubject) } from 'rxjs';

import {sources} from 'eventsourcemock';

jest.setTimeout(1400000);

it('TC\_PROG\_0001 Verify DTAL subscribe() receives and publishes SSE', async () => {

  try {

//create RxJS subject

const subject = new [ReplaySubject](https://rxjs.dev/api/index/class/ReplaySubject)(3);

//create a map of event vs subject

let eventRxJSMap<string, [ReplaySubject](https://rxjs.dev/api/index/class/ReplaySubject)> = new Map()

eventRxJSMap.set("pacemaker",subject)

//create instance of DTAL (unit under test)

let DTALinstance = new DTAL(eventRxJSMap)

//test notification payload for pacemarker

const messageEvent = new MessageEvent('pacemarker',

{"thingId":"pacemaker:00-01-02-03-04-05","features":{"deviceparameters":{"properties":{"MAIN\_PROCESSOR\_DOWNLOADS\_PERFORMED":49}},"batteryparameters":{"properties":{"BLUETOOTH\_MESSAGES\_TRIM":200}},"leadimpedancestream":{"properties":{"raw":"a5 a5 a5 a5 0b 0e eb ff 00 00 a8 5b 5a 5a 5a 5a","leadImpedanceValues":"25853.956521739132,28338.428571428574"}}},"\_revision":1388,"\_modified":"2023-03-14T15:01:43.228341376Z"}

);

subject.subscribe({

//this callback should be called by DTAL.subscribe method

//verify the received payload matches messageEvent

})

DTALinstance.subscribe("pacemaker")

sources["https://evaluate.samarthii.com/api/2/things?ids=pacemaker"].onmessage(

messageEvent.type,

messageEvent

);

  } catch (e) { expect(e).toMatch('error'); }

});