EventBus javascript library

User Guide

[EventBus Library 3](#_Toc505090145)

[EventBus API Reference 3](#_Toc505090146)

[Usage 4](#_Toc505090147)

[Tests 4](#_Toc505090148)

# EventBus Library

This Library allows to publish actions, and manage the action handler definition for this action. This library was developed with javascipt.

* 1. Library structure

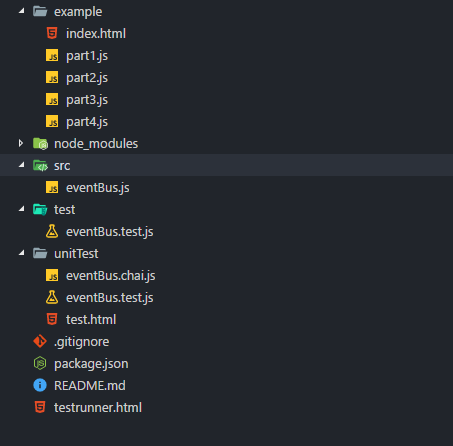


Figure – library structure

# EventBus API Reference

**On**: use to add action inside action handler

*//* @action - string

*//* @actionHandler - function

eventBus.on(action, actionHandler);

**Emit**: use to broacast actions

*//* @action - string

eventBus.emit(action);

**Remove:** use to remove the action and action handler from the broker

*//* @action - string

eventBus.remove(action);

# Usage

To use this library you need to download/clone it from Gitlab : [git@gitlab.com:TOFA/eventBus.git](mailto:git@gitlab.com:TOFA/eventBus.git) and import the script()inside the html file as following :

<script type="text/javascript" language="javascript" src="../src/eventBus.js"></script>

In the js file eventBus need to initialize the library as following:

*//* init eventBus lib

var eventBus = new EventBus();

To register action with action handler you need to use ‘on’ api as following:

*//* register action with action handler

eventBus.on('one', function () {

console.log('one');});

To perform registered action you need to use ‘emit’ api as following:

*//* emit action (perform action)

eventBus.emit('one');

To remove registered action in the broker you need to use ‘remove’ api as following:

*//* remove action

eventBus.remove('one');

**Full example:**

var eventBus = new EventBus();

*//*---------------------------------- We should call one handler when an action is fired

eventBus.on('one', function () {

console.log('one');

});

eventBus.emit('one');

For more example you can open index.html in example folder and check in the browser console the results of the 4 javascript file present in example folder.

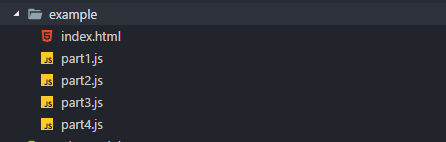


Figure example folder

# Tests

1. Unit tests without framework

To check the unit test in unitTest folder open test.html and check that the tests passed are green

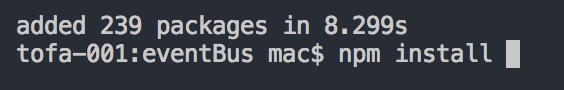


Figure - unit test of eventBus

1. Unit tests with chai.js

To check the unit test with chaiJs and mochaJs you need to run npm install in the root of the lib folder and after open testrunner.html

nodeJs must be installed in your machine and run npm install in the root of the folder using the terminal



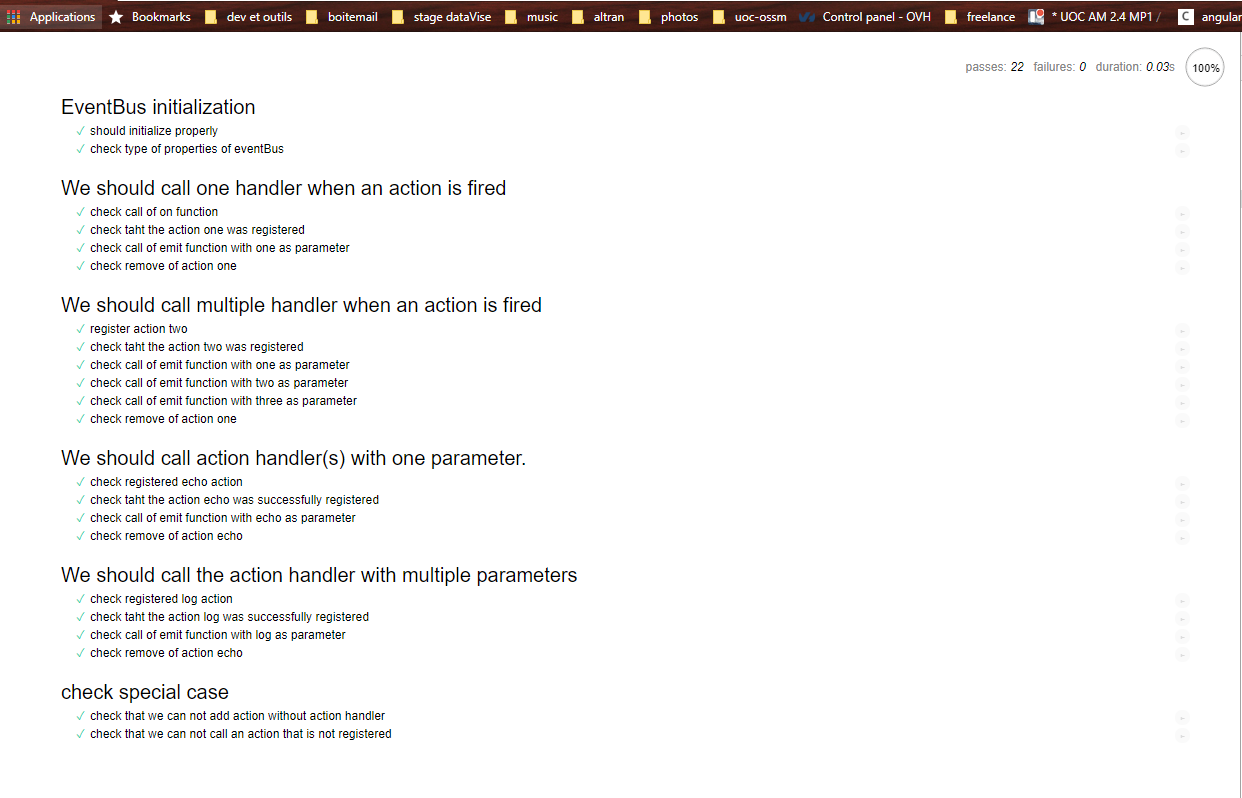


Figure unit test with chaiJs and mochaJs

To have more details for every test please click on one of test suite title or subtitle.

1. Unit test coverage

**To check the unit test coverage, you need to open you terminal in the root of folder library and run this command (nodeJs must be installed in your machine).**

npm run test-with-coverage

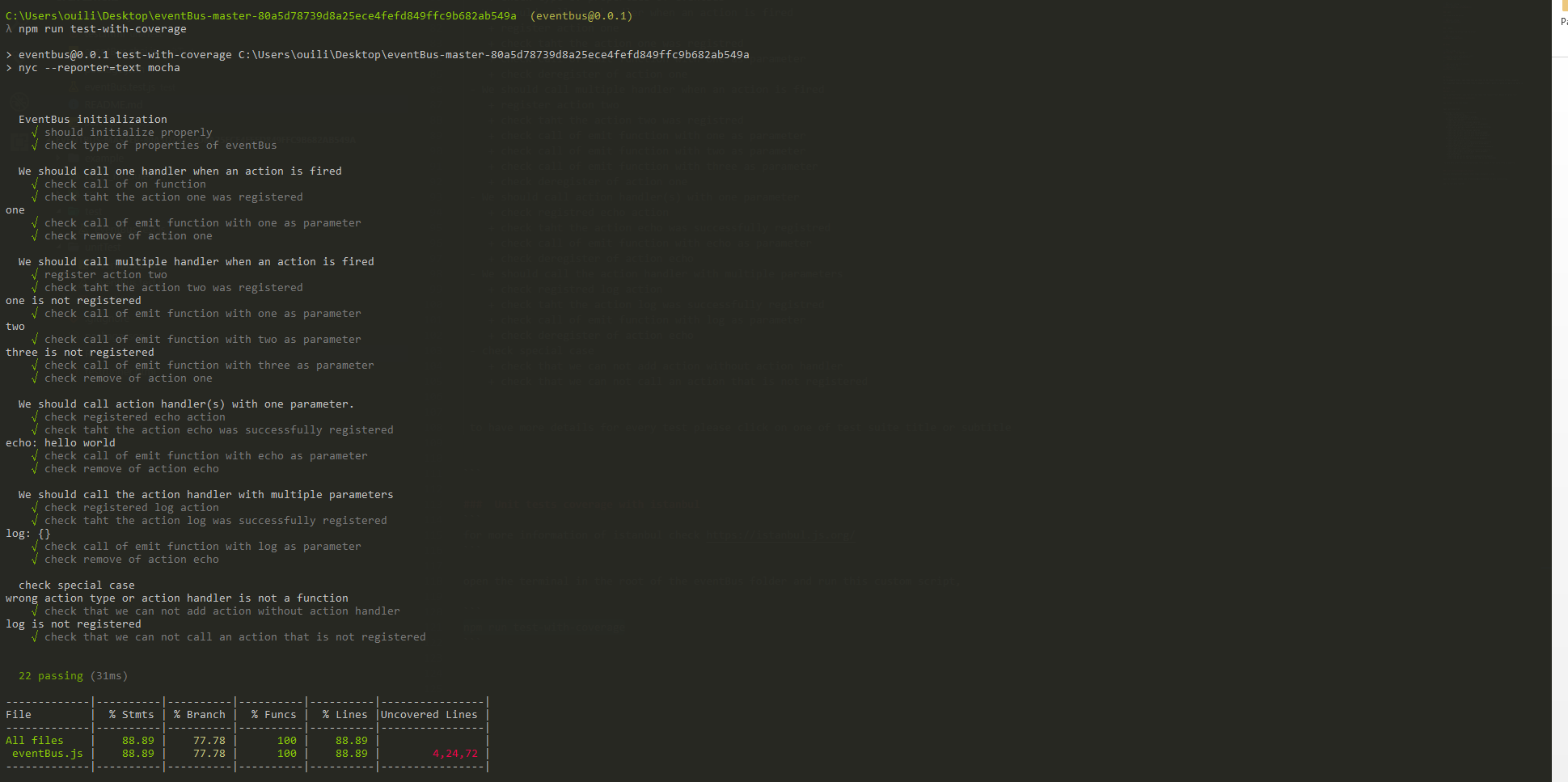
****

Figure - unit test coverage