


Annotation 7

Title: Back End Testing Framework

Description: This artifact is a collection of screenshots from ElectronChat project source code; specifically the testing environment and devDependencies. I was in charge of setting up the testing framework and importing the libraries used for the unit tests. The unit tests was written and managed by all members of ElectronChat team (James Marshall, Philip Marshall, Timothy Trusov, and Zakhaddin Khalidov). In the software development life cycle, testing is one of the most important and overlooked efforts of the project. Active involvement and promotion of testing is required for source code to be properly tested. This becomes most apparent when the people that wrote the original code are not the people maintaining it, therefor not having direct knowledge of how the code should be working. We made it an effort to test as much of our source code as possible by implementing mochajs and chaijs testing libraries, to further promote good coding habits.

Reflection: By implementing a testing framework that resulted in this article I gained insight into the most efficient testing practices implemented in NodeJs. Although I have experience in unit testing other platforms such as Python, I learned how to use the chaijs library to create assertions and unit test a NodeJs back end using the mocha test framework. While setting up the testing environment, I ran across a common occurrence for software developers. In many cases other developers identify weak points of a library and build other libraries to fulfill tasks that are not available. In this case the assertion library provided by mocha was crude and did not flow as other simpler testing frameworks. I decided to implement the chaijs assertion library because it reads more like plain English. And if in the future anyone else looks at my unit tests they will be easier to comprehend and hence making the tests user friendly.

Artifact 7

back-end-main > test >  DbConnectTest.js > ...

```
1  const assert = require('chai').assert;
2  const DbConnect = require('../app/Models/DbConnect');
3  const interface = require('../app/Models/SqlInterface')
4  describe('DbConnector', function() {
5      const ip = '127.0.0.1';
6      const port = '8080';
7      let inter = new interface(port,ip);
8      let connector = new DbConnect(inter);
9
10     describe("getPort", function() {
11         it('getPort should return 8080', function(){
12             assert.equal(connector.getPort(), '8080');
13         })
14     })
15
16     describe("getIp", function() {
17         it('getIP should return 127.0.0.1', function(){
18             assert.equal(connector.getIp(), '127.0.0.1');
19         })
20     })
21
22     describe("getAllMsgs", function() {
23         it('getALLMsgs should return true', function(){
24             assert.equal(connector.getAllMsgs("Hello_world"), true);
25         })
26     })
27
28     describe("getMsg", function() {
29         it('getMsgs should return true', function(){
30             assert.equal(connector.getMsg(), true);
31         })
32     })
33 }
```

```
14  "devDependencies": {
15      "chai": "^4.3.6",
16      "codecov": "^3.8.3",
17      "jsdoc": "^4.0.0",
18      "mocha": "^10.0.0",
19      "nodemon": "^2.0.20",
20      "nyc": "^15.1.0"
21  },
```