

Lesson 11 Demo 03

Implementing Asynchronous and Synchronous Testing with Mocha

Objective: To perform asynchronous and synchronous testing for a Node.js sample app using Mocha for robust validation of code functionality and execution scenarios

Tools Required: Visual Studio

Prerequisites: Knowledge of JavaScript, synchronous/asynchronous concept, and Node.js

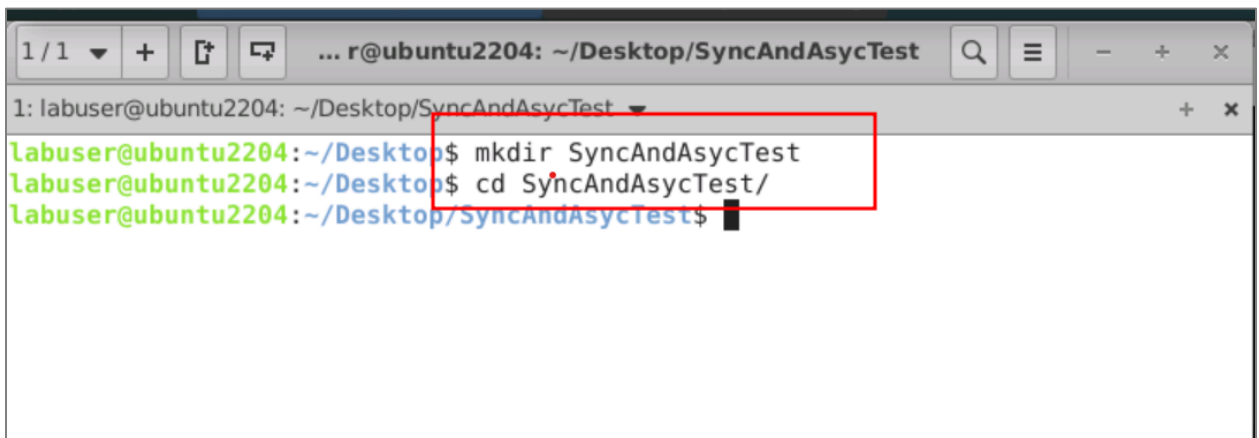
Steps to be followed:

1. Create the project structure, package.json file, and install Mocha module
2. Create a simple Node.js program and test file to test synchronous and asynchronous code

Step 1: Create the project structure, package.json file, and install Mocha module

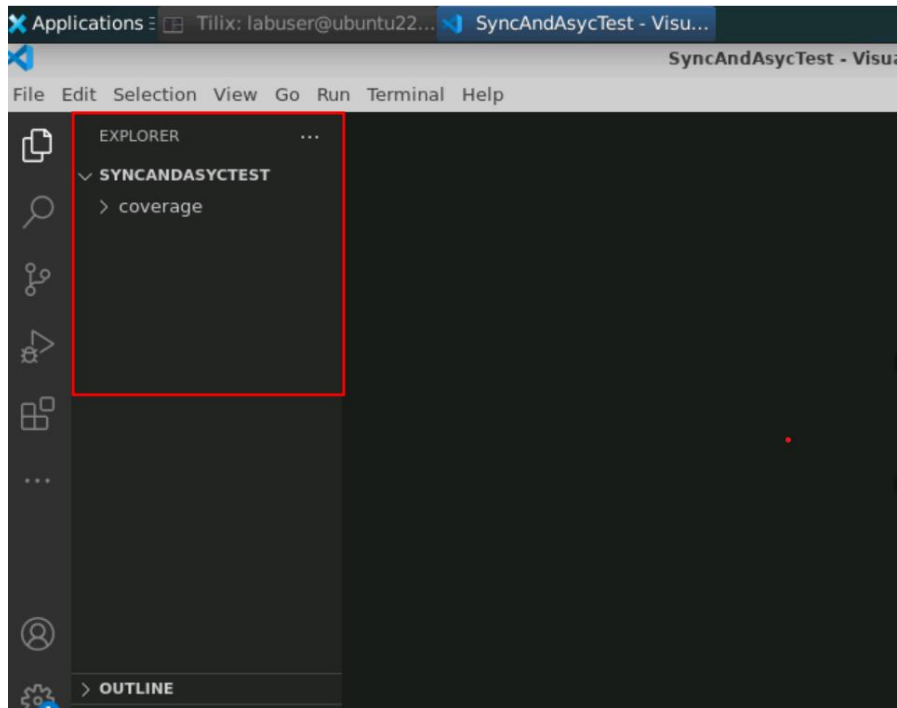
- 1.1 Run the following code to create a directory to hold the application and make it the working directory:

```
mkdir SyncAndAsyncTest
cd SyncAndAsyncTest
```



```
1 / 1  +  ... r@ubuntu2204: ~/Desktop/SyncAndAsyncTest  Q  -  +  x
1: labuser@ubuntu2204: ~/Desktop/SyncAndAsyncTest
labuser@ubuntu2204:~/Desktop$ mkdir SyncAndAsyncTest
labuser@ubuntu2204:~/Desktop$ cd SyncAndAsyncTest/
labuser@ubuntu2204:~/Desktop/SyncAndAsyncTest$
```

1.2 Open this folder in VSCode IDE and open the SyncAndAsyncTest folder



1.3 Open the terminal and run the following command to create a **package.json** file for the application

npm init -y

```
1: labuser@ubuntu2204: ~/Desktop/SyncAndAsyncTest
labuser@ubuntu2204:~/Desktop$ mkdir SyncAndAsyncTest
labuser@ubuntu2204:~/Desktop$ cd SyncAndAsyncTest/
labuser@ubuntu2204:~/Desktop/SyncAndAsyncTest$ npm init -y
Wrote to /home/labuser/Desktop/SyncAndAsyncTest/package.json:

{
  "name": "syncandasycctest",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

labuser@ubuntu2204:~/Desktop/SyncAndAsyncTest$
```

This command creates the package.json file

Now, you can view the package.json file

```

() package.json x
{} package.json > ...
1  {
2    "name": "nodejstestingmocha",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1"
8    },
9    "author": "",
10   "license": "ISC"
11 }
12

```

1.4 Install **npm install mocha -D** using the following command:

```

1: labuser@ubuntu2204: ~/Desktop/SyncAndAsyncTest
labuser@ubuntu2204: ~/Desktop/SyncAndAsyncTest$ npm install mocha -D
added 77 packages, and audited 78 packages in 6s
20 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
labuser@ubuntu2204:~/Desktop/SyncAndAsyncTest$

```

You will be able to view these dependencies in package.json file

```

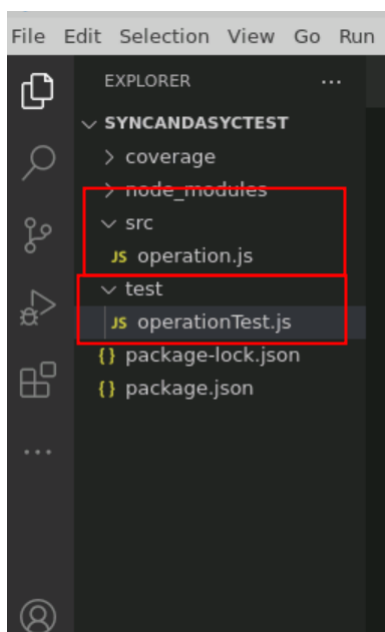
() package.json x
{} package.json > {} devDependencies > mocha
1  {
2    "name": "nodejstestingmocha",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1"
8    },
9    "author": "",
10   "license": "ISC",
11   "devDependencies": {
12     "mocha": "^10.2.0"
13   }
14 }
15

```

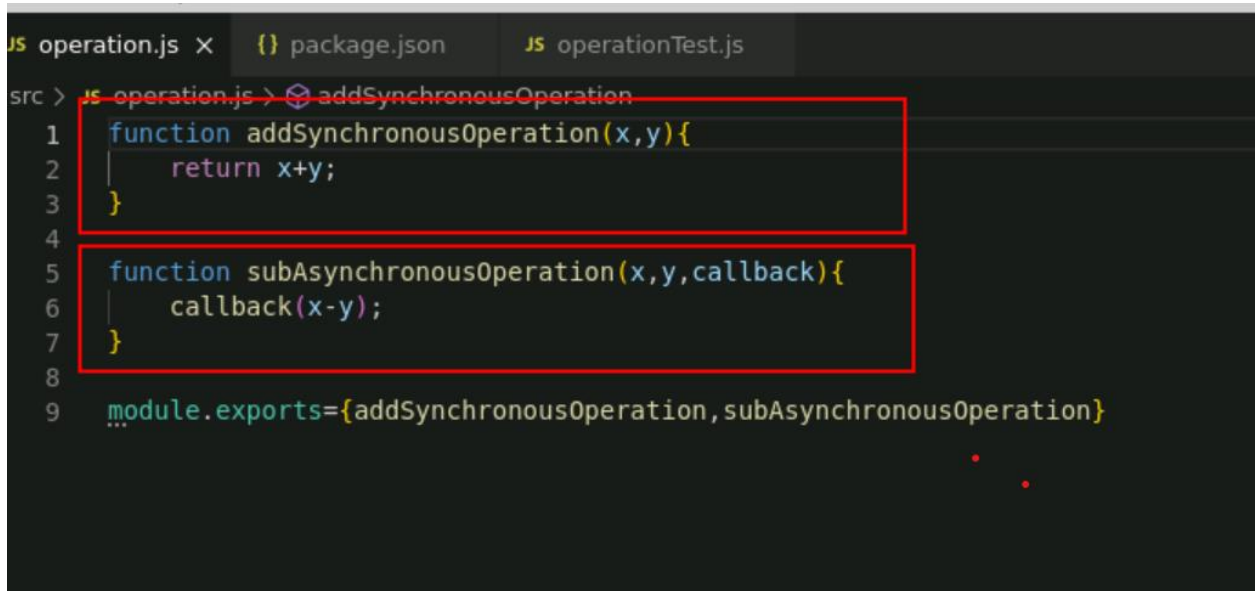
Step 2: Create a simple Node.js program and test file to test synchronous and asynchronous code

- 2.1 Create a src folder which contains all the source code. Inside the src folder, create operation.js
- 2.2 Create a test folder which contains all the source code. Inside the test folder, create operationTest.js

After creating all the folders and files, view the project structure



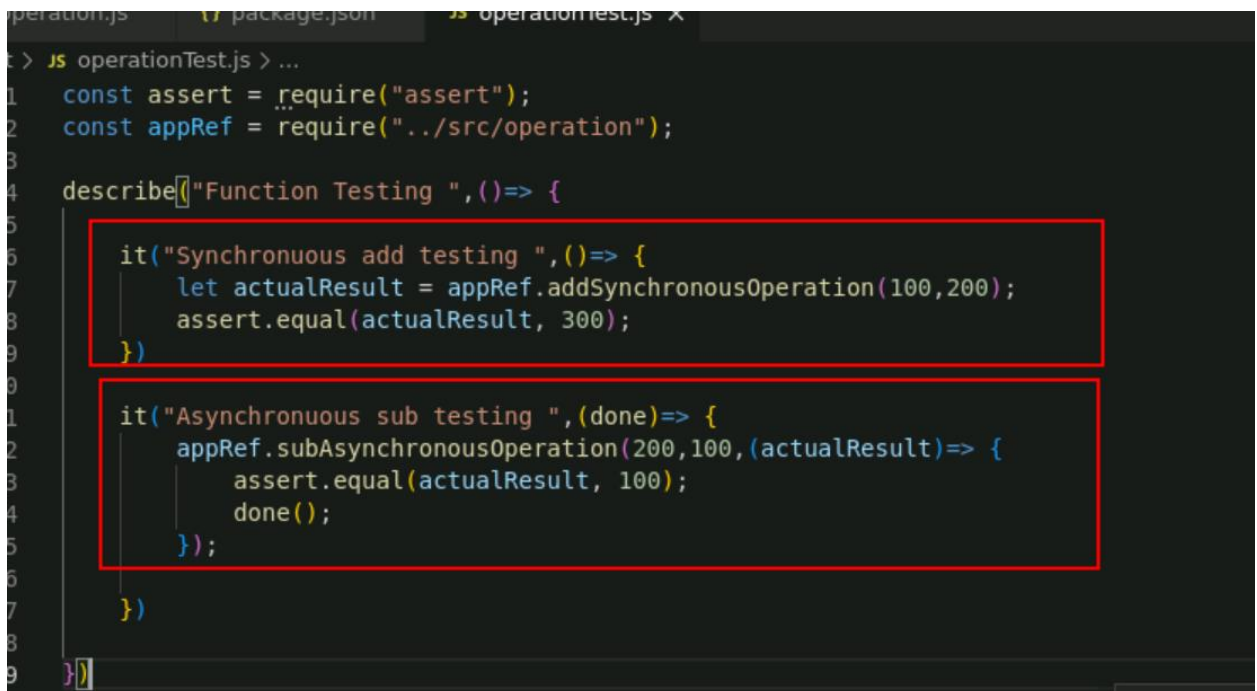
2.3 Create a simple **operation.js** file with the following code as shown in the screenshot below:



```

src > JS operation.js > addSynchronousOperation
1 function addSynchronousOperation(x,y){
2   return x+y;
3 }
4
5 function subAsynchronousOperation(x,y,callback){
6   callback(x-y);
7 }
8
9 module.exports={addSynchronousOperation,subAsynchronousOperation}
  
```

2.4 Create a simple **operationTest.js** file with the following code as shown in the screenshot below:



```

operation.js  package.json  JS operationTest.js
src > JS operationTest.js > ...
1 const assert = require("assert");
2 const appRef = require("../src/operation");
3
4 describe("Function Testing ",()=> {
5
6   it("Synchronous add testing ",()=> {
7     let actualResult = appRef.addSynchronousOperation(100,200);
8     assert.equal(actualResult, 300);
9   })
10
11   it("Asynchronous sub testing ",(done)=> {
12     appRef.subAsynchronousOperation(200,100,(actualResult)=> {
13       assert.equal(actualResult, 100);
14       done();
15     });
16   })
17 })
18
19
  
```

2.5 In the **package.json** file, provide the testing file details

```
{ } package.json > { } devDependencies
1  {
2    "name": "syncandasycctest",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "test": "mocha test/*.js"
8    },
9    "keywords": [],
10   "author": "",
11   "license": "ISC",
12   "devDependencies": {
13     "mocha": "^10.2.0"
14   }
15 }
16
```

2.6 Run the following command in the **terminal** to run the file:
npm test

```
1/1  +  ... r@ubuntu2204: ~/Desktop/SyncAndAsyncTest
1: labuser@ubuntu2204: ~/Desktop/SyncAndAsyncTest
labuser@ubuntu2204: ~/Desktop/SyncAndAsyncTest$ npm test

> syncandasycctest@1.0.0 test
> mocha test/*.js

Function Testing
  ✓ Synchronous add testing
  ✓ Asynchronous sub testing

2 passing (122ms)

labuser@ubuntu2204:~/Desktop/SyncAndAsyncTest$
```

2.7 Make the changes in **operationTest.js** file to make the test fail

```
est > js operationTest.js > describe("Function Testing ") callback
1  const assert = require("assert");
2  const appRef = require("../src/operation");
3
4  describe("Function Testing ",()=> {
5
6      it("Synchronous add testing ",()=> {
7          let actualResult = appRef.addSynchronousOperation(100,200);
8          assert.equal(actualResult, 200);
9      })
10
11     it("Asynchronous sub testing ",(done)=> {
12         appRef.subAsynchronousOperation(200,100,(actualResult)=> {
13             assert.equal(actualResult, 200);
14             done();
15         });
16     })
17 })
18
19 })
```

Now run the test again to make it fail

```
labuser@ubuntu2204: ~/Desktop/SyncAndAsyncTest$ npm test

> syncandasycTest@1.0.0 test
> mocha test/*.js

Function Testing
  1) Synchronous add testing
  2) Asynchronous sub testing

0 passing (48ms)
2 failing

1) Function Testing
   Synchronous add testing :

   AssertionError [ERR_ASSERTION]: 300 == 200
   + expected - actual

   -300
   +200
```

```
at Context.<anonymous> (test/operationTest.js:8:16)  
at process.processImmediate (node:internal/timers:471:21)
```

2) Function Testing

Asynchronous sub testing :

```
AssertionError [ERR_ASSERTION]: 100 == 200  
+ expected - actual
```

```
-100  
+200
```

```
at /home/labuser/Desktop/SyncAndAsyncTest/test/operationTest.js:13:20  
at Object.subAsynchronousOperation (src/operation.js:6:5)  
at Context.<anonymous> (test/operationTest.js:12:16)  
at process.processImmediate (node:internal/timers:471:21)
```

By following these steps, you have successfully implemented asynchronous and synchronous testing for a Node.js sample app using Mocha, ensuring thorough validation of code functionality and execution scenarios.