

Lesson 11 Demo 06

Implementing Fetch API with Multiple Done()

Objective: To install node-fetch-commonJS for making REST API calls and testing that API using Mocha in a Node.js environment

Tools Required: Visual Studio

Prerequisites: Knowledge of JavaScript, Promise, Node.js, and REST API

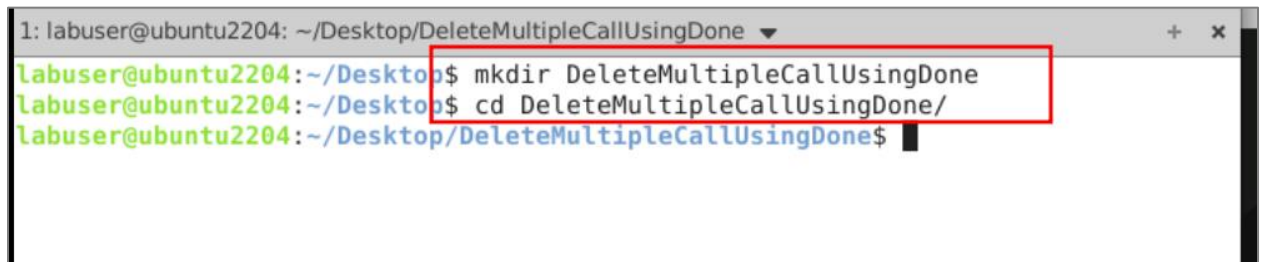
Steps to be followed:

1. Create the node app.js file which calls the third-party fake REST API node-fetch-commonJS module
2. Create a test file to check the REST API using done

Step 1: Create the Node app.js file which calls the third-party fake REST API using node-fetch-commonJS module

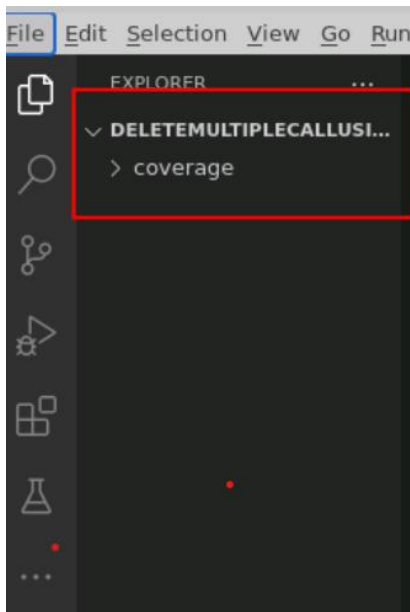
- 1.1 Run the following code to create a directory to hold the application and make it the working directory:

```
mkdir DetectMultipleCallUsingDone  
cd DetectMultipleCallUsingDone
```

A terminal window screenshot showing the execution of two commands. The first command is 'mkdir DetectMultipleCallUsingDone' and the second is 'cd DetectMultipleCallUsingDone/'. The second command and its output are highlighted with a red rectangle. The terminal title bar shows the path '~/Desktop/DeleteMultipleCallUsingDone'.

```
1: labuser@ubuntu2204: ~/Desktop/DeleteMultipleCallUsingDone  
labuser@ubuntu2204:~/Desktop$ mkdir DetectMultipleCallUsingDone  
labuser@ubuntu2204:~/Desktop$ cd DetectMultipleCallUsingDone/  
labuser@ubuntu2204:~/Desktop/DeleteMultipleCallUsingDone$
```

1.2 In the VSCode IDE, open the folder as shown in the screenshot below:



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

```
labuser@ubuntu2204:~/Desktop/DeleteMultipleCallUsingDone$ npm init -y
Wrote to /home/labuser/Desktop/DeleteMultipleCallUsingDone/package.json:

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

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

It creates a package.json file with default option

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 Mocha using **npm install mocha -D** command

```

1: labuser@ubuntu2204: ~/Desktop/DeleteMultipleCallUsingDone
labuser@ubuntu2204:~/Desktop/DeleteMultipleCallUsingDone$ npm install mocha -D

up to date, audited 78 packages in 825ms

20 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
labuser@ubuntu2204:~/Desktop/DeleteMultipleCallUsingDone$

```

1.5 Run **npm install node-fetch-commonjs -D** command

```

found 0 vulnerabilities
labuser@ubuntu2204:~/Desktop/DeleteMultipleCallUsingDone$ npm install node-fetch
-commonjs -D

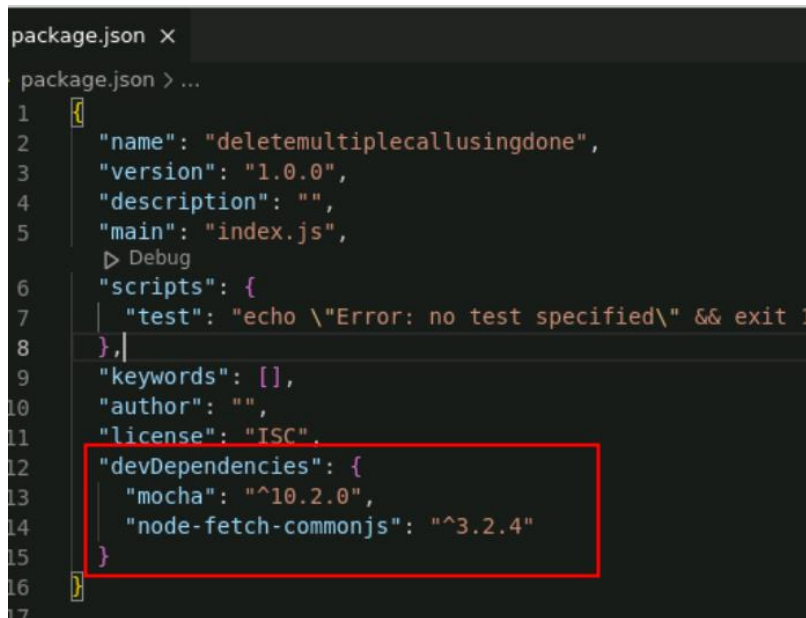
added 5 packages, and audited 83 packages in 3s

23 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
labuser@ubuntu2204:~/Desktop/DeleteMultipleCallUsingDone$

```

You can view these dependencies in the package.json file

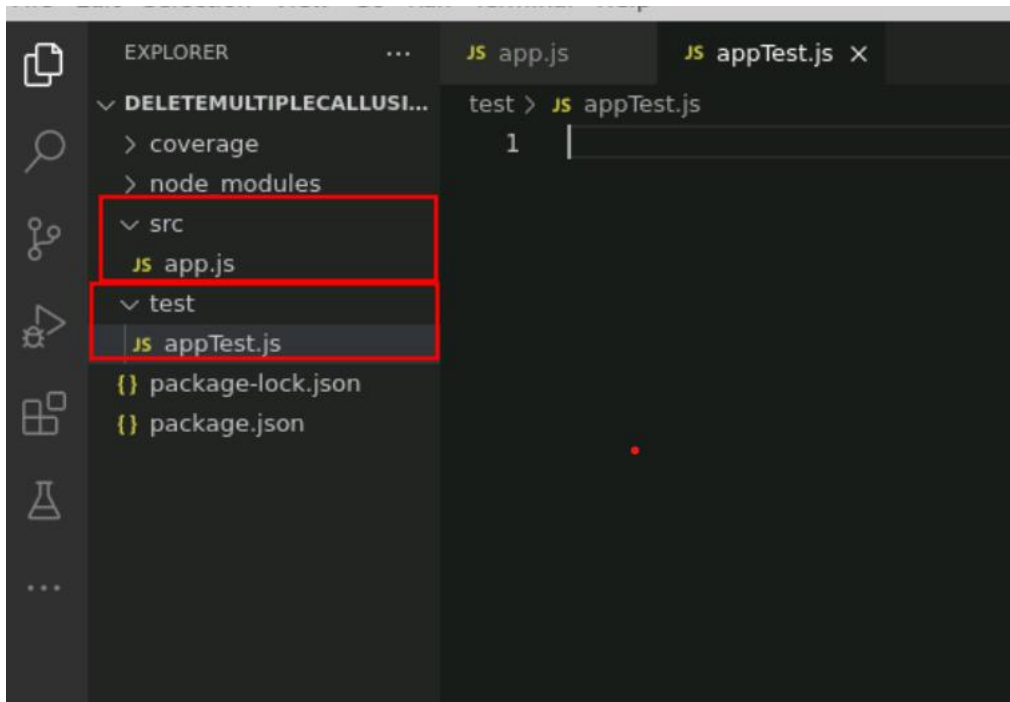


```
package.json x
package.json > ...
1 {
2   "name": "deletemultiplecallusingdone",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "test": "echo \"Error: no test specified\" && exit 1",
8   },
9   "keywords": [],
10  "author": "",
11  "license": "ISC",
12  "devDependencies": {
13    "mocha": "^10.2.0",
14    "node-fetch-commonjs": "^3.2.4"
15  }
16 }
```

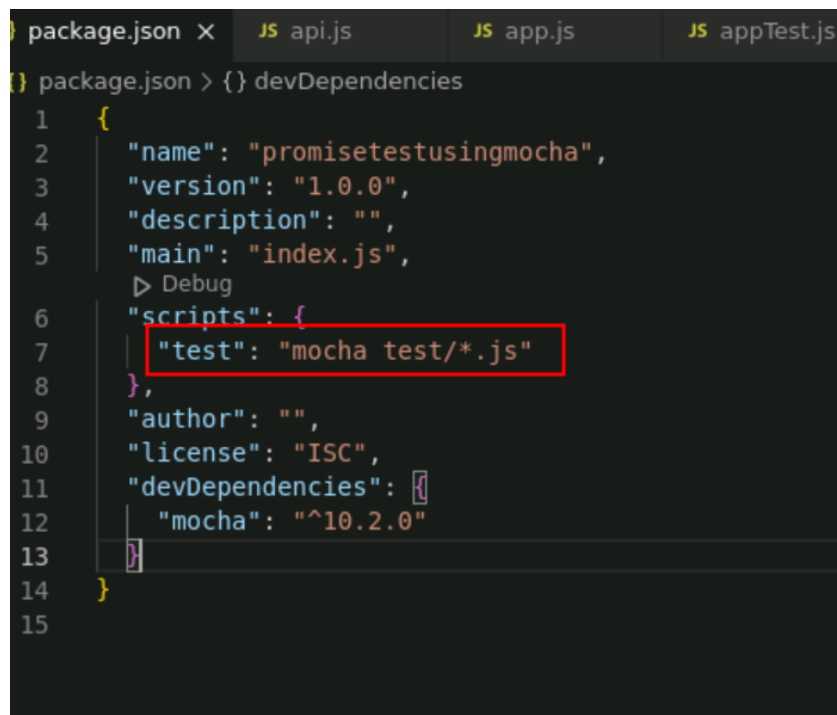
Step 2: Create a test file to check the REST API using done

- 2.1 Create a src folder which contains all source code and inside the src folder, create app.js
- 2.2 Create a test folder which contains all testing files and inside the test folder, create appTest.js

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

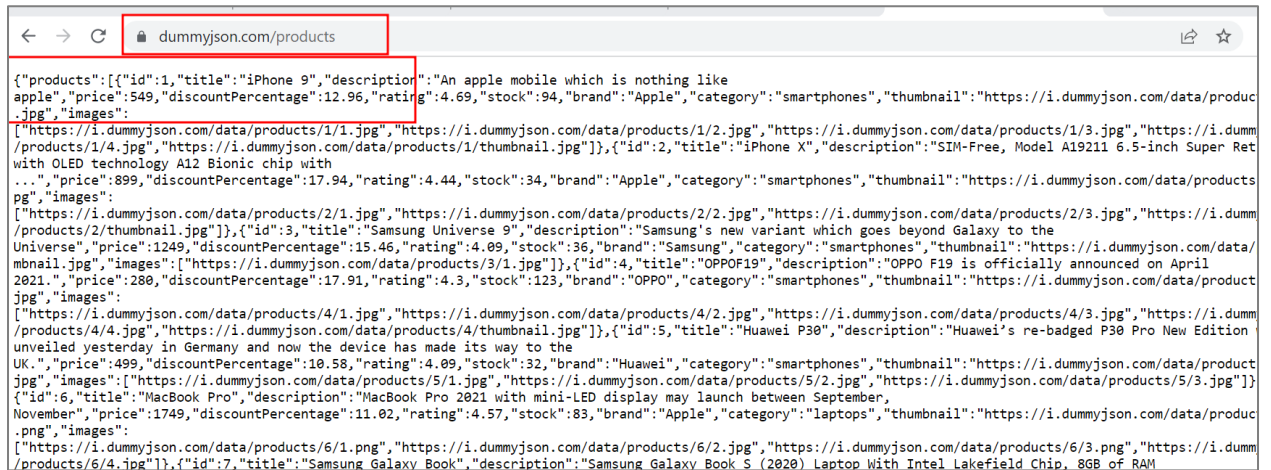


2.3 Now, make the following changes in the package.json file

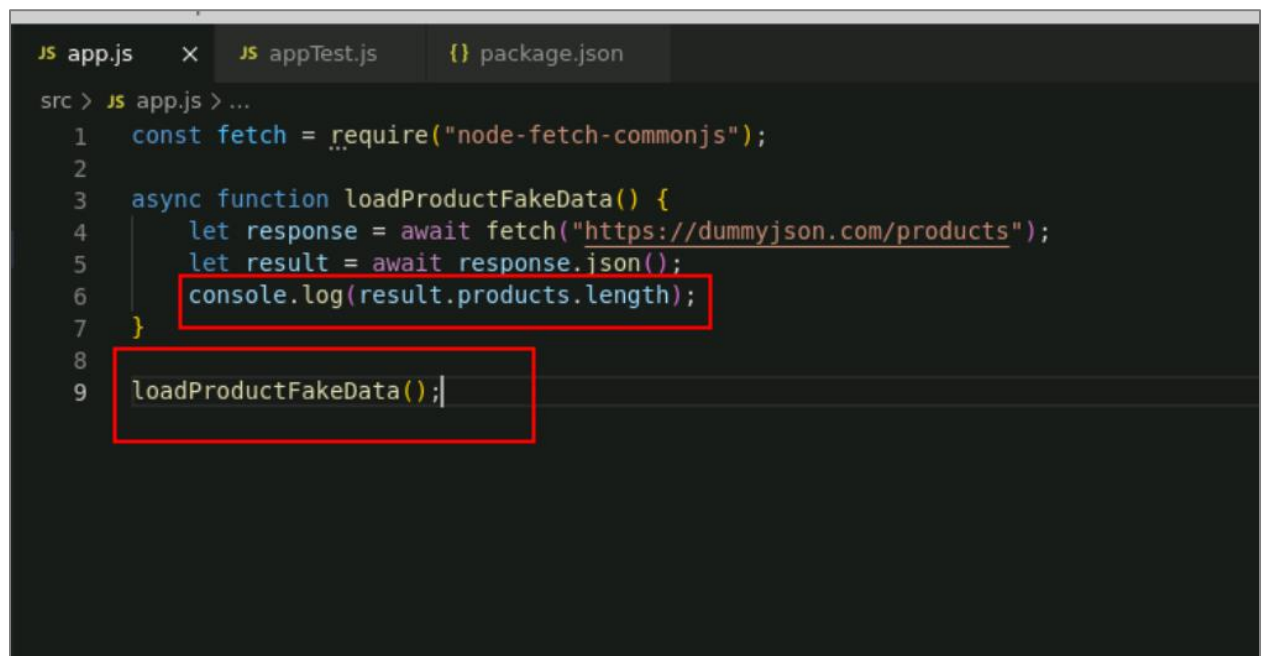


2.4 First, check for fake rest api which gives a collection of project details in the json format.

<https://dummyjson.com/products>

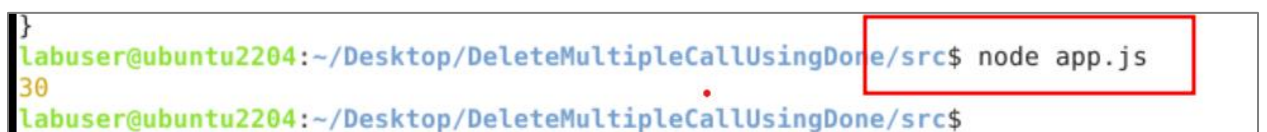


2.5 Call the rest api with the help of a third-party module, that is, node-fetch-commonjs module



Before the test, please check whether this program is working properly or not.

Open the command prompt inside a src folder and run this application



Now, make the changes in the app.js file

```

JS app.js  X  JS appTest.js  {} package.json
src > JS app.js > [?] <unknown>
1  const fetch = require("node-fetch-commonjs");
2
3  async function loadProductFakeData() {
4      let response = await fetch("https://dummyjson.com/products");
5      let result = await response.json();
6      //console.log(result.products.length);
7      return result;
8  }
9
10 //loadProductFakeData();
11
12 module.exports={loadProductFakeData}

```

2.6 Write the code to test rest api in appTest.js

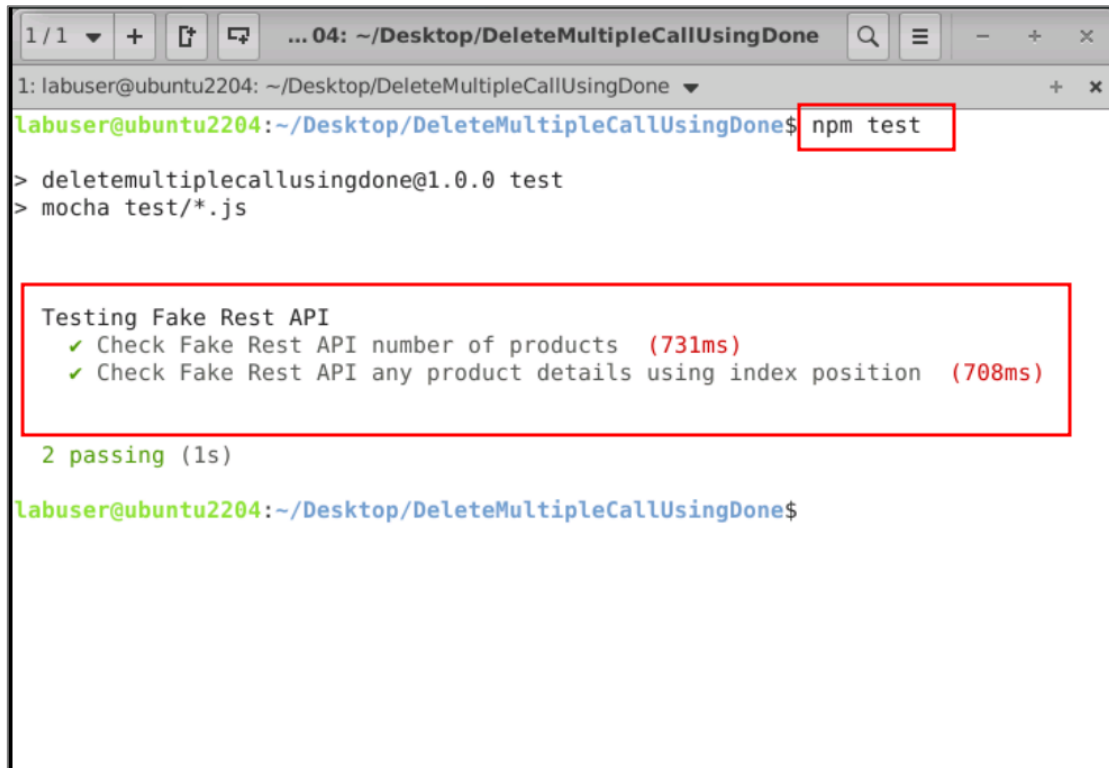
```

> JS appTest.js > describe("Promise Fake Test API") callback
const assert = require("assert");
const appRef = require("../src/app");
describe("Promise Fake Test API", ()=> {
    it("Product found test ", async ()=> {
        const actulaResult = await appRef.findProduct(1);
        const expectedResult = {
            "message": "Product found",
            "result": {
                pid: 1,
                pname: "Tv",
                price: 55000
            }
        };
        assert.deepEqual(actulaResult, expectedResult);
    })

    it("Product not found test ", async ()=> {
        try{
            const actulaResult = await appRef.findProduct(5);
        }catch(error){
            assert.equal(error.message, "Product not found");
        }
    })
})

```

2.7 Run the following command in the **terminal** to run the file: (make sure you are inside a project but outside a src folder. Here you are running test file).



```

1 / 1 + [ ] ... 04: ~/Desktop/DeleteMultipleCallUsingDone
1: labuser@ubuntu2204: ~/Desktop/DeleteMultipleCallUsingDone
labuser@ubuntu2204:~/Desktop/DeleteMultipleCallUsingDone$ npm test

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

Testing Fake Rest API
  ✓ Check Fake Rest API number of products (731ms)
  ✓ Check Fake Rest API any product details using index position (708ms)

2 passing (1s)

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

Here npm test checks the **package.json** file and all testing file details are already provided in test mocha test/*.js



```

package.json x JS api.js JS app.js JS appTest.js
package.json > {} devDependencies
1 {
2   "name": "promisetestingmocha",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "test": "mocha test/*.js"
8   },
9   "author": "",
10  "license": "ISC",
11  "devDependencies": {
12    "mocha": "^10.2.0"
13  }
14 }
15
  
```

By following these steps, you have successfully installed node-fetch-commonJS for making REST API calls and tested the API functionality using Mocha in a Node.js environment to ensure robust integration and verification.