

Lesson 11 Demo 10

Implementing Express.js App Testing

Objective: To create REST APIs using Express.js and demonstrate thorough testing with Mocha and SuperTest for robust validation and verification

Tools Required: Visual Studio

Prerequisites: Knowledge of JavaScript, Node.js, Express.js and RESTful web service

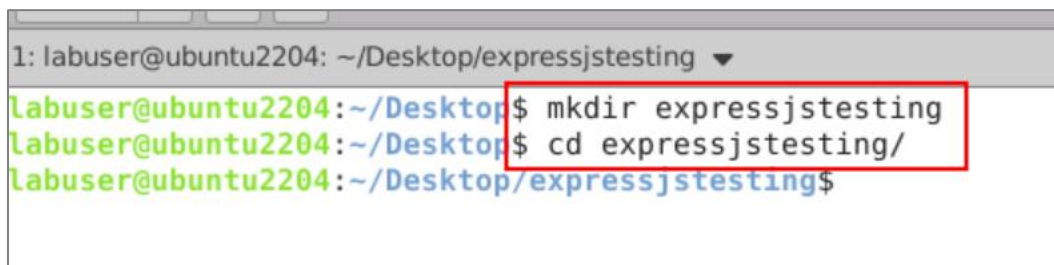
Steps to be followed:

1. Create the project structure, package.json file and install mocha and SuperTest party module
2. Create the REST API using Express.js
3. Test the REST APIs using Mocha with SuperTest

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

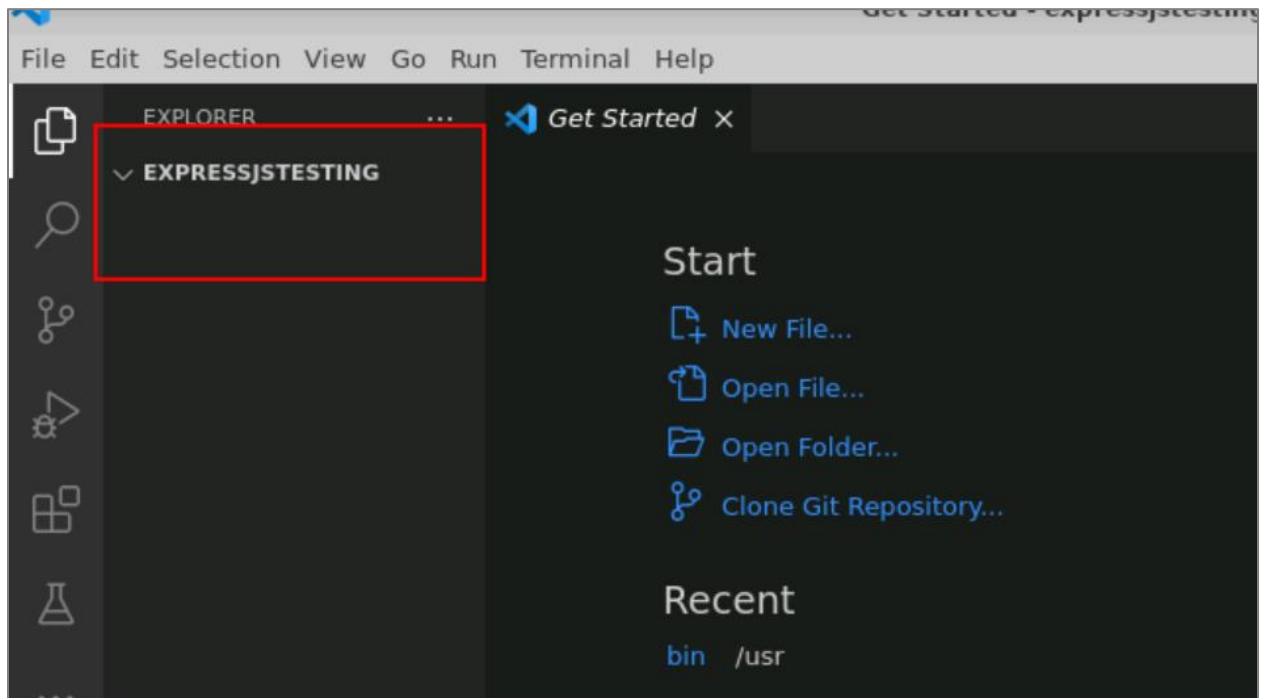
- 1.1 Run the below code to create a working directory to hold the application:

```
mkdir expressjstesting  
cd expressjstesting
```



```
1: labuser@ubuntu2204: ~/Desktop/expressjstesting ▼  
labuser@ubuntu2204:~/Desktop$ mkdir expressjstesting  
labuser@ubuntu2204:~/Desktop$ cd expressjstesting/  
labuser@ubuntu2204:~/Desktop/expressjstesting$
```

1.2 Open this folder in VSCode IDE



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

npm init -y

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

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

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

It creates a package.json file with a default option.

You can now 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 Express.js by using **npm install express**; Install Mocha Supertest by using **npm install mocha supertest -D**.

```

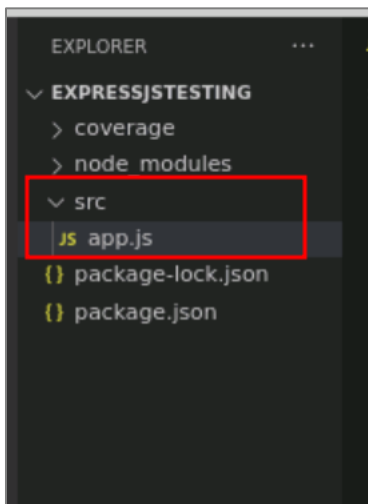
labuser@ubuntu2204:~/Desktop/expressjstesting$ npm install express
up to date, audited 58 packages in 812ms
7 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
labuser@ubuntu2204:~/Desktop/expressjstesting$ npm install mocha supertest -D
added 94 packages, and audited 152 packages in 3s
27 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
labuser@ubuntu2204:~/Desktop/expressjstesting$

```

Now you can see the package.json file.

Step 2: Create the REST API using Express.js

2.1 Create an src folder, and create an app.js file inside it

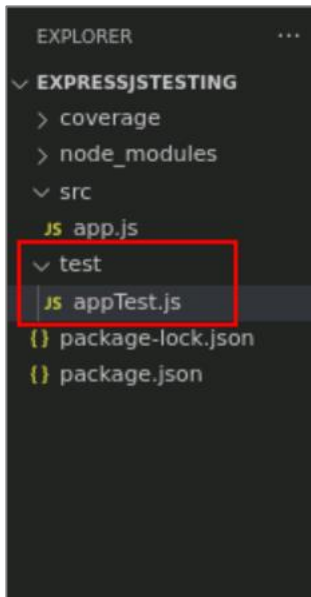


2.2 Create the below code in the app.js file, which contains the REST API

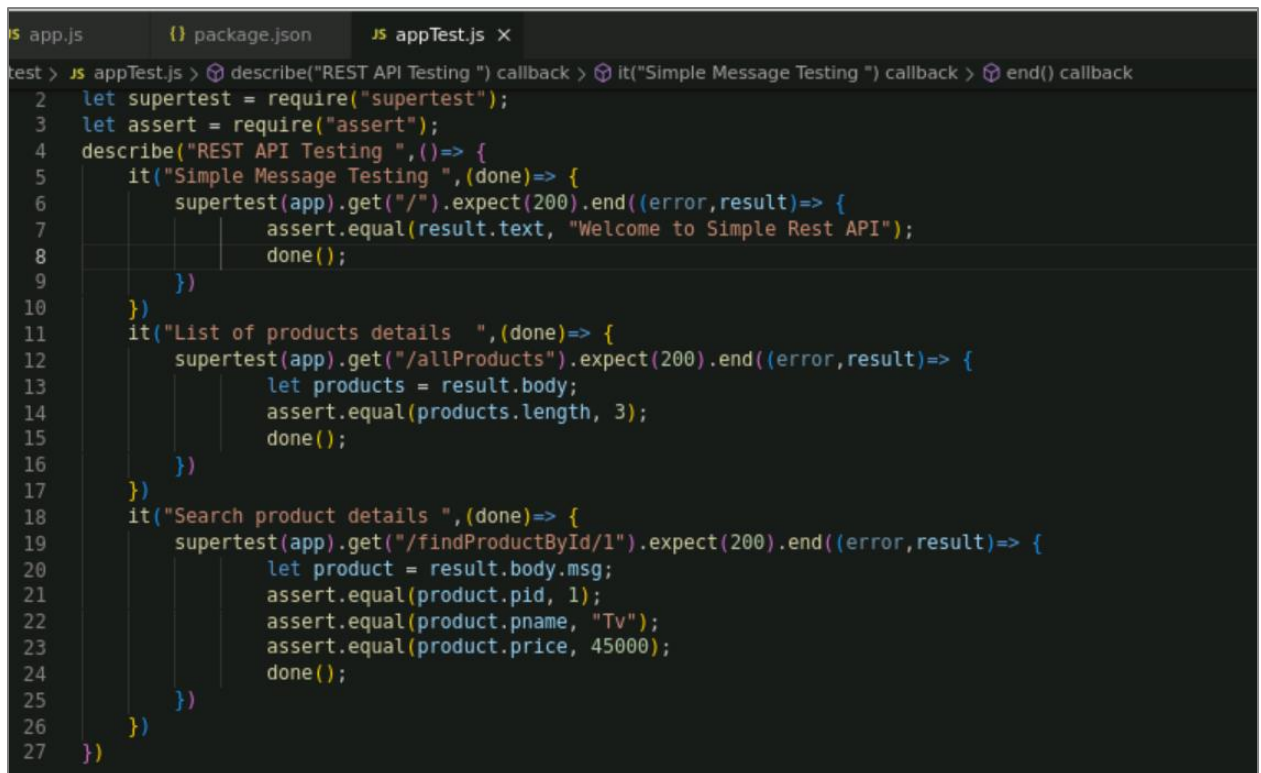
```
js app.js > [0] <unknown>
let express = require("express");
let app = express();
let products=[
  {pid:1,pname:"Tv",price:45000},
  {pid:2,pname:"Computer",price:34000},
  {pid:3,pname:"Laptop",price:87000}
]
app.get("/",(request,response)=> {
  response.send("Welcome to Simple Rest API")
})
app.get("/allProducts",(request,reseonse)=> {
  reseonse.json(products);
})
app.get("/findProductById/:pid",(request,response)=> {
  let pid = request.params.pid;
  let result = products.find(p=>p.pid==pid);
  if(result==undefined){
    response.json({"msg":"Product record not present"})
  }else {
    response.json({"msg":result});
  }
})
app.listen(3000,()=>console.log("Server running on port number 3000"));
module.exports=app;
```

Step 3: Test the REST API using Mocha with SuperTest

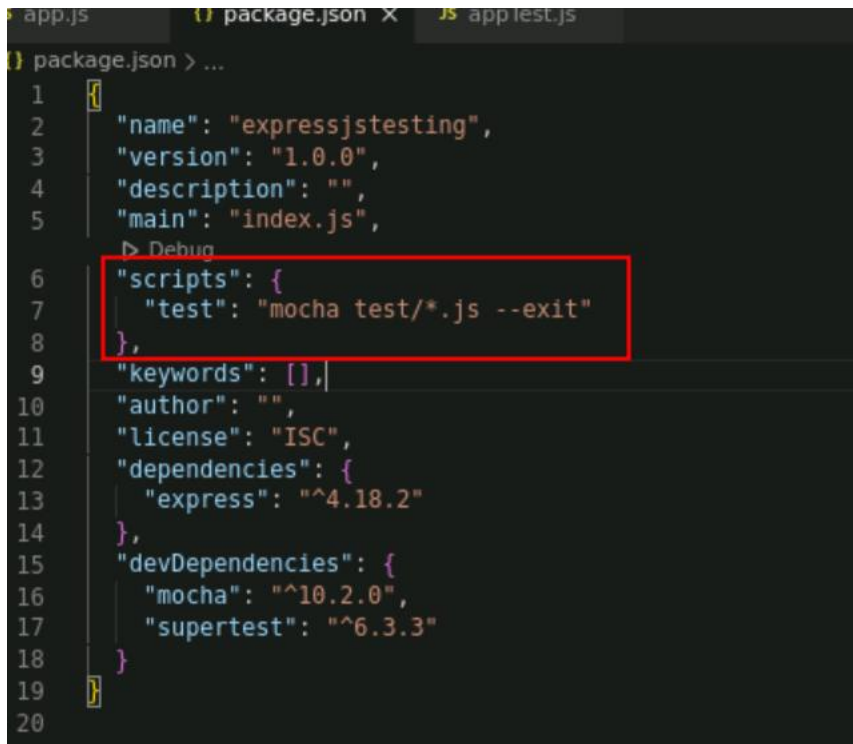
3.1 Create a test folder, and create an appTest.js file inside it



3.2 Add the code in appTest.js as shown in the screenshot below:



3.3 Now provide the testing file details in the package.json file



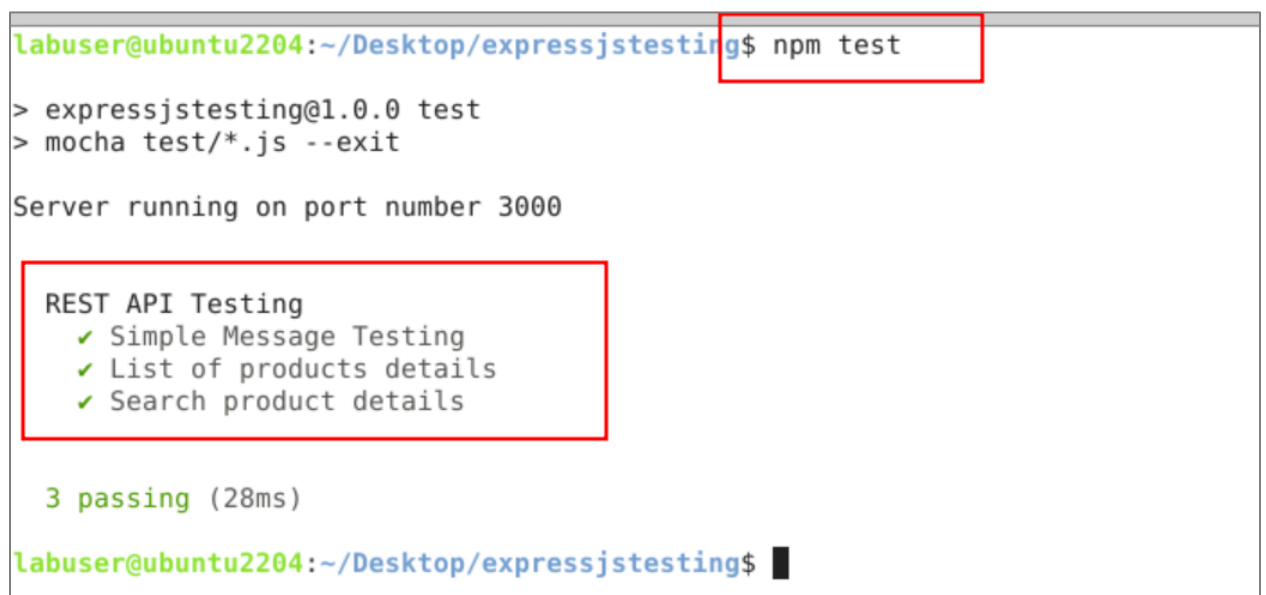
```

1  {
2    "name": "expressjstesting",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "test": "mocha test/*.js --exit"
8    },
9    "keywords": [],
10   "author": "",
11   "license": "ISC",
12   "dependencies": {
13     "express": "^4.18.2"
14   },
15   "devDependencies": {
16     "mocha": "^10.2.0",
17     "supertest": "^6.3.3"
18   }
19 }
20

```

3.4 Evaluate the code using the below command:

npm test



```

labuser@ubuntu2204:~/Desktop/expressjstesting$ npm test

> expressjstesting@1.0.0 test
> mocha test/*.js --exit

Server running on port number 3000

REST API Testing
  ✓ Simple Message Testing
  ✓ List of products details
  ✓ Search product details

3 passing (28ms)

labuser@ubuntu2204:~/Desktop/expressjstesting$

```

By following these steps, you have successfully created REST APIs using Express.js and conducted comprehensive testing with Mocha and SuperTest to ensure robust validation and verification of API functionality.