

Lesson 11 Demo 02

Demonstrating Expression and Arrow Style Testing in a Node.js Sample App with Mocha

Objective: To perform testing with expression and arrow style for a Node.js sample app using Mocha for validating functionality and syntax

Tools Required: Visual Studio

Prerequisites: Knowledge of JavaScript and Node.js

Steps to be followed:

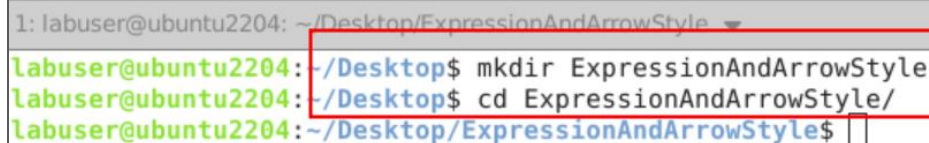
1. Create project structure, package.json file and installing Mocha module
2. Create simple Node.js program and test file

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

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

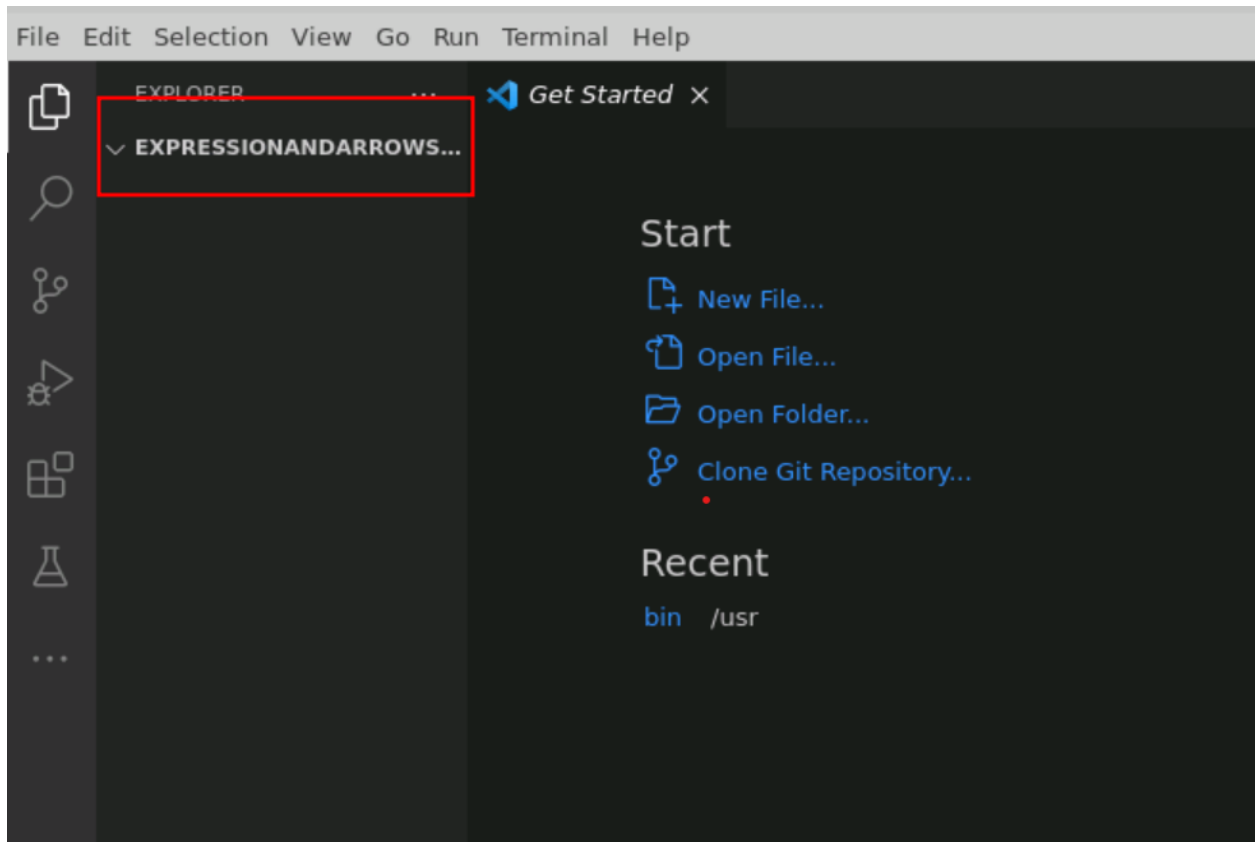
mkdir ExpressionAndArrowStyle

cd ExpressionAndArrowStyle

A terminal window screenshot showing the execution of two commands. The first command is 'mkdir ExpressionAndArrowStyle' and the second is 'cd ExpressionAndArrowStyle/'. The prompt is 'labuser@ubuntu2204: ~/Desktop\$'. A red rectangle highlights the two command lines.

```
labuser@ubuntu2204: ~/Desktop$ mkdir ExpressionAndArrowStyle
labuser@ubuntu2204: ~/Desktop$ cd ExpressionAndArrowStyle/
labuser@ubuntu2204: ~/Desktop/ExpressionAndArrowStyle$
```

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**

```
labuser@ubuntu2204:~/Desktop/ExpressionAndArrowStyle$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (expressionandarrowstyle)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to /home/labuser/Desktop/ExpressionAndArrowStyle/package.json:
```

Note: Just enter for all questions and keep default values and Enter Yes at last



```

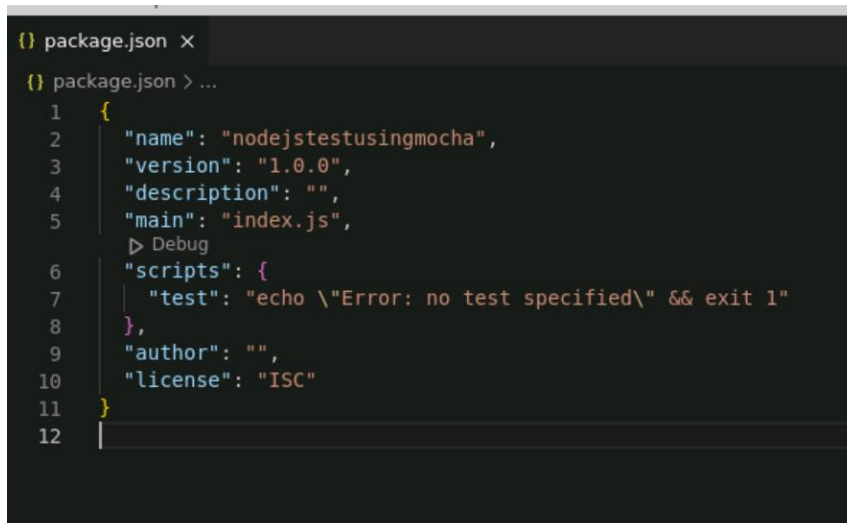
1 / 1 + ... ntu2204: ~/Desktop/NodeJSTestUsingMocha
1: labuser@ubuntu2204: ~/Desktop/NodeJSTestUsingMocha
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to /home/labuser/Desktop/NodeJSTestUsingMocha/package.json:

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

Is this OK? (yes) yes
File Manager
Browse the file system

```

You can now view the package.json file



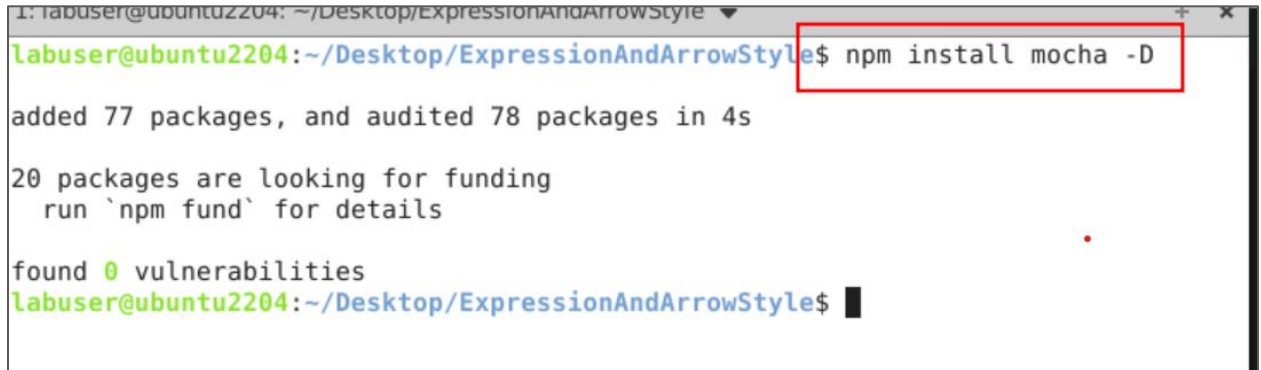
```

{} 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 **npm install mocha -D** using the following command:

(-D means testing dependencies required only in development mode.)

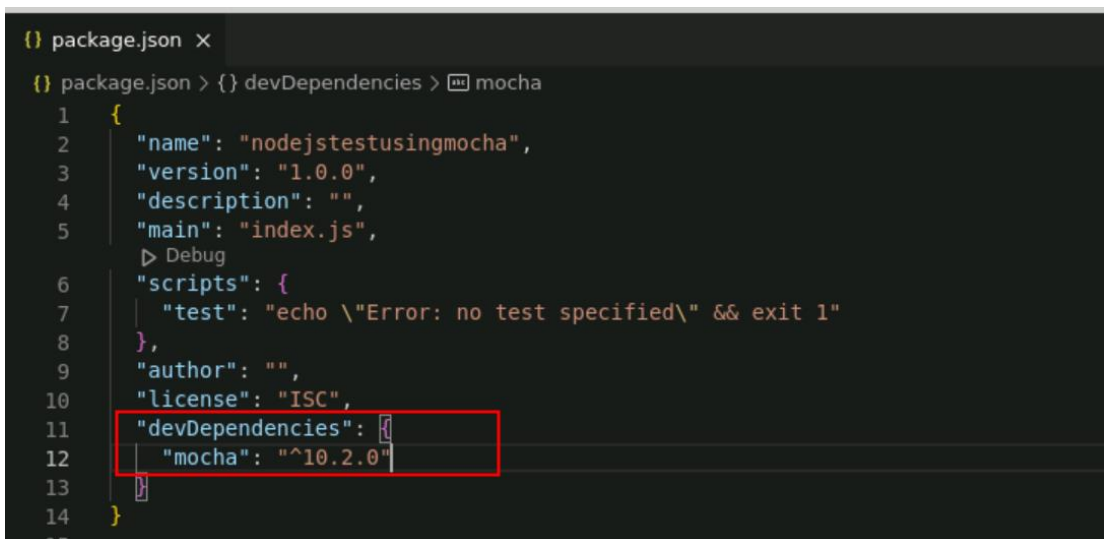


```
labuser@ubuntu2204: ~/Desktop/ExpressionAndArrowStyle
labuser@ubuntu2204:~/Desktop/ExpressionAndArrowStyle$ npm install mocha -D
added 77 packages, and audited 78 packages in 4s

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

found 0 vulnerabilities
labuser@ubuntu2204:~/Desktop/ExpressionAndArrowStyle$
```

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



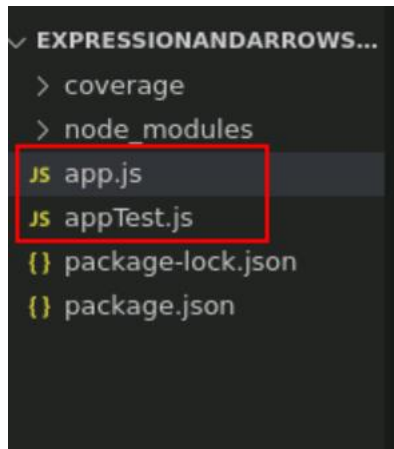
```
{
  "name": "nodejstestusingmocha",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "mocha": "^10.2.0"
  }
}
```

Step 2: Create simple Node.js Program and test file

2.1 Create the following files:

app.js, that contains JavaScript functions

appTest.js, that contains Mocha testing code



2.2 Create a simple app.js file with below code:

```
app.js > ...
1  function sayHello(name){
2      |   return "Welcome to node js mocha "+name;
3  }
4
5
6
7  module.exports = {sayHello}
8
9
```

2.3 Create a simple appTest.js file with below code:

```
appTest.js > ...
1  const assert = require("assert");
2  const appRef = require("./app");
3
4  // expression style
5  describe("Simple Testing using expression style",function(){
6      it("sayHello testing using expression style",function(){
7          let actualResult = appRef.sayHello("John");
8          assert.equal(actualResult,"Welcome to node js mocha John");
9      })
10 })
11
12 // arrow style
13 describe("Simple Testing using arrow style",()=>{
14     it("sayHello testing using expression style",()=>{
15         let actualResult = appRef.sayHello("John");
16         assert.equal(actualResult,"Welcome to node js mocha John");
17     })
18 })
19
```

2.4 Run the below command in the **terminal** to run the file:

```
labuser@ubuntu2204: ~/Desktop/ExpressionAndArrowStyle
labuser@ubuntu2204:~/Desktop/ExpressionAndArrowStyle$ mocha appTest.js

Simple Testing using expression style
  ✓ sayHello testing using expression style

Simple Testing using arrow style
  ✓ sayHello testing using expression style

2 passing (6ms)

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

By following these steps, you have successfully implemented testing with expression and arrow style in a Node.js sample app using Mocha to ensure thorough validation of functionality and syntax.