Lesson 08 Demo 02

Handling GET and POST requests

Objective: To demonstrate GET and POST requests for an Express.js application to ensure functionality and response verification

Tools Required: Visual Studio, Postman, Express and, Node.js

Prerequisites: Knowledge of JavaScript and Node.js

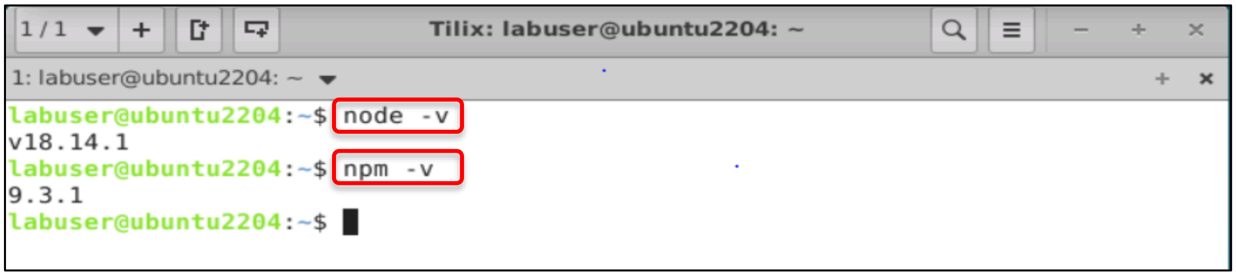
Steps to be followed:

1. Verify the Node.js installation
2. Create an Express app
3. Install and configure Postman
4. Handle GET and POST requests through Postman

Step 1: Verify the Node.js installation

1.1 Verify the Node.js installation using the following commands:

node -v
npm -v

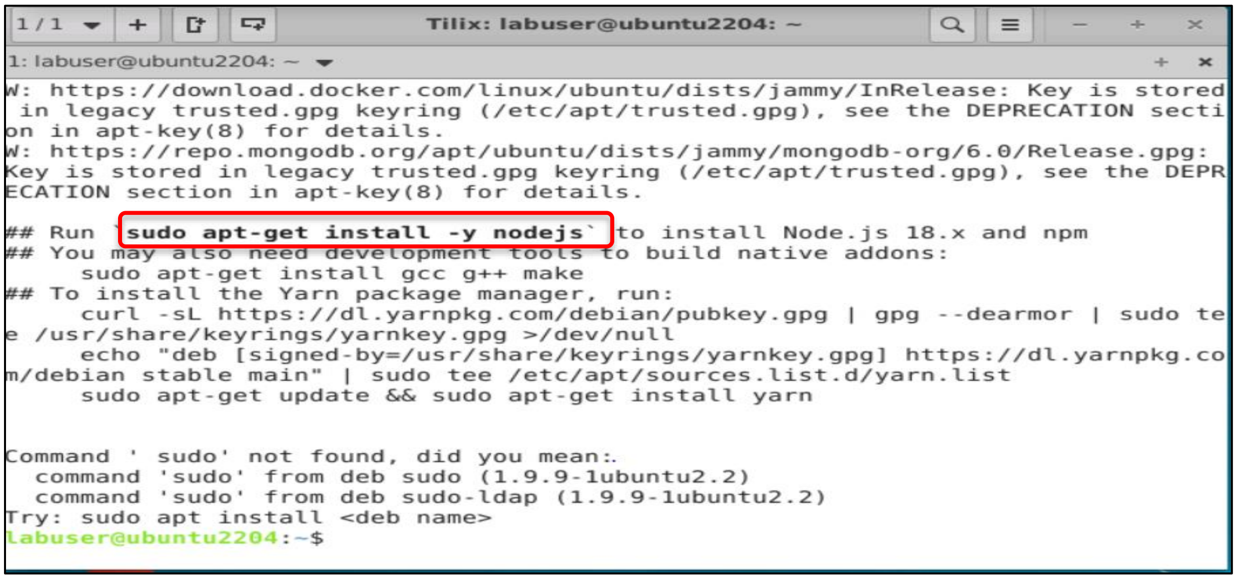


```
labuser@ubuntu2204:~$ node -v
v18.14.1
labuser@ubuntu2204:~$ npm -v
9.3.1
labuser@ubuntu2204:~$
```

Note: If the Node.js installation is successful, you will see the above output

1.2 Run the below command to install Node.js (if it is not installed on your system).

sudo apt-get install -y Node.js


Tilix: labuser@ubuntu2204: ~

1: labuser@ubuntu2204: ~

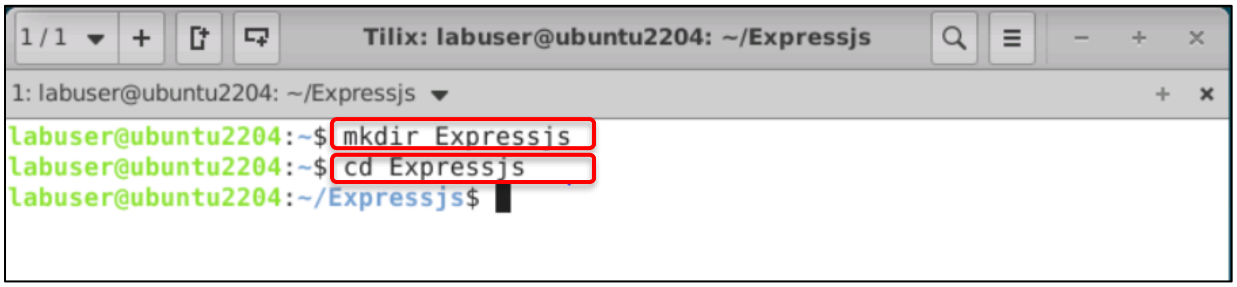
```
W: https://download.docker.com/linux/ubuntu/dists/jammy/InRelease: Key is stored
 in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION secti
on in apt-key(8) for details.
W: https://repo.mongodb.org/apt/ubuntu/dists/jammy/mongodb-org/6.0/Release.gpg:
Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPR
ECATION section in apt-key(8) for details.

## Run `sudo apt-get install -y nodejs` to install Node.js 18.x and npm
## You may also need development tools to build native addons:
     sudo apt-get install gcc g++ make
## To install the Yarn package manager, run:
     curl -sL https://dl.yarnpkg.com/debian/pubkey.gpg | gpg --dearmor | sudo te
e /usr/share/keyrings/yarnkey.gpg >/dev/null
     echo "deb [signed-by=/usr/share/keyrings/yarnkey.gpg] https://dl.yarnpkg.co
m/debian stable main" | sudo tee /etc/apt/sources.list.d/yarn.list
     sudo apt-get update && sudo apt-get install yarn


Command ' sudo' not found, did you mean:
  command 'sudo' from deb sudo (1.9.9-1ubuntu2.2)
  command 'sudo' from deb sudo-ldap (1.9.9-1ubuntu2.2)
Try: sudo apt install <deb name>
labuser@ubuntu2204:~$
```

Step 2: Create an Express application

2.1 Create a directory to hold the application using the commands given below and make it the working directory:

mkdir Expressjs

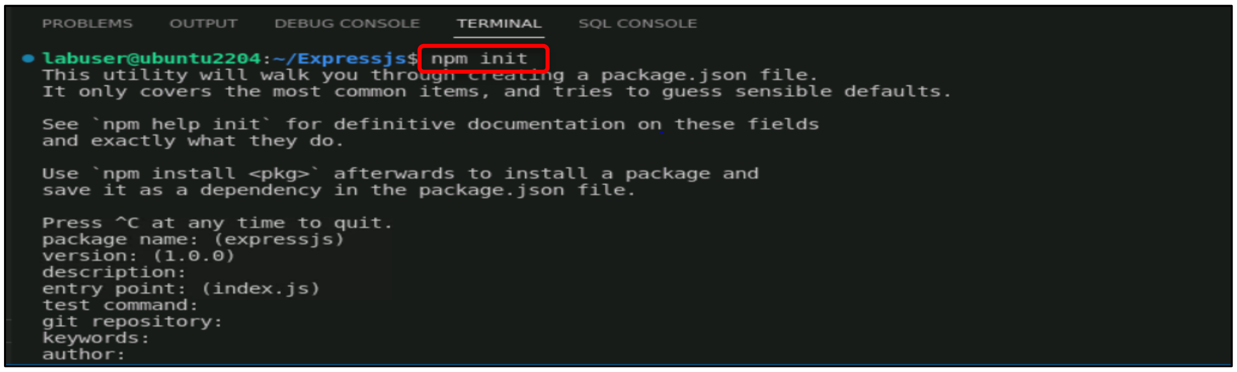
cd Expressjs


Tilix: labuser@ubuntu2204: ~/Expressjs

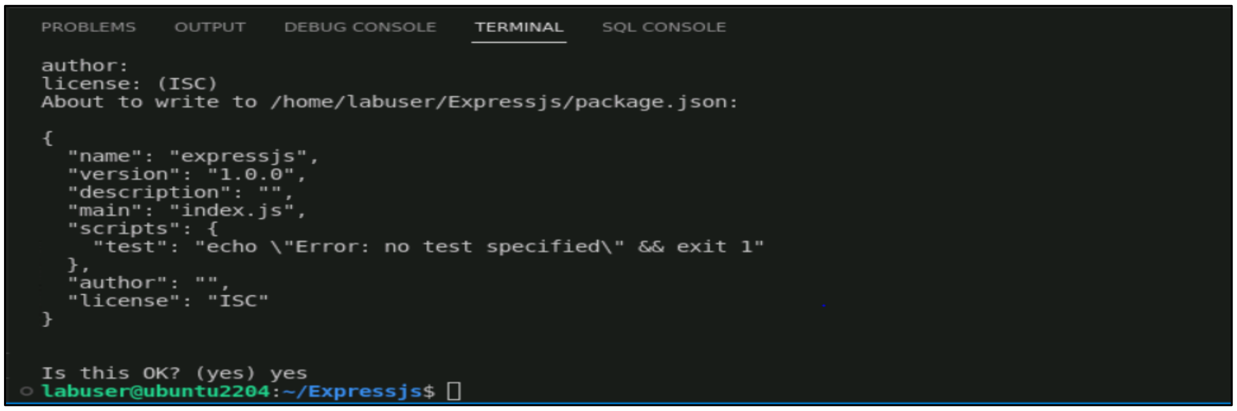
1: labuser@ubuntu2204: ~/Expressjs

```
labuser@ubuntu2204:~$ mkdir Expressjs
labuser@ubuntu2204:~$ cd Expressjs
labuser@ubuntu2204:~/Expressjs$
```

2.2 Open the working directory in VS Code. Now, open the terminal and create a **package.json** file by executing the following command:
npm init

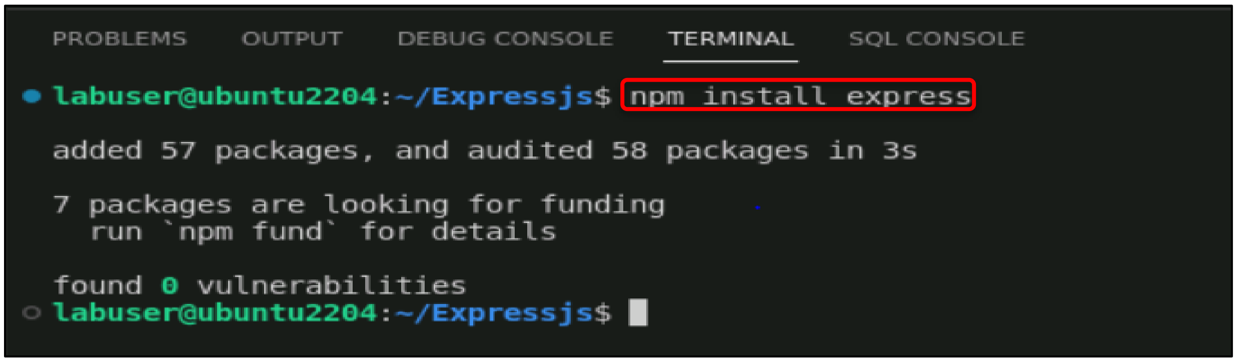


```
labuser@ubuntu2204:~/Expressjs$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

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: (expressjs)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
```



```
author:
license: (ISC)
About to write to /home/labuser/Expressjs/package.json:

{
  "name": "expressjs",
  "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
labuser@ubuntu2204:~/Expressjs$
```

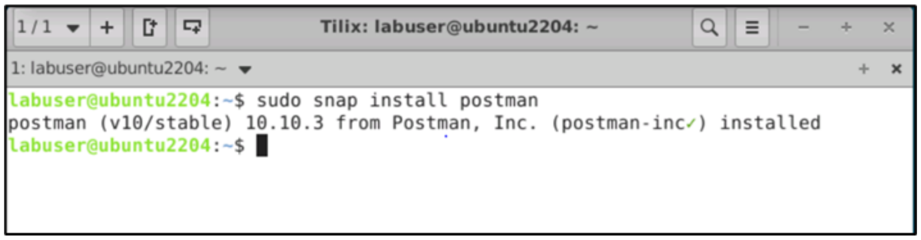
2.3 Install Express.js in the **Expressjs** directory using the following command and save it in the dependencies list:
npm install express



```
labuser@ubuntu2204:~/Expressjs$ npm install express

added 57 packages, and audited 58 packages in 3s

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

found 0 vulnerabilities
labuser@ubuntu2204:~/Expressjs$
```

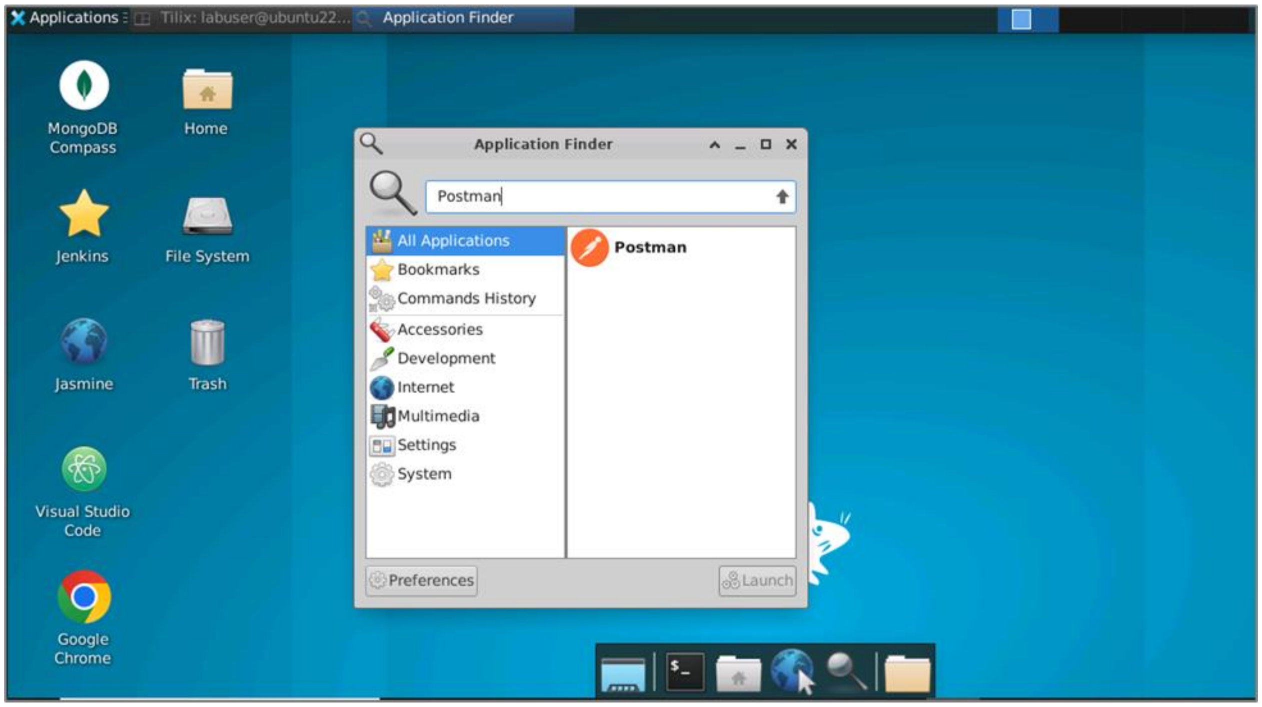
Step 3: Install and configure Postman

3.1 Run the following command in the system terminal to install Postman:
sudo snap install postman

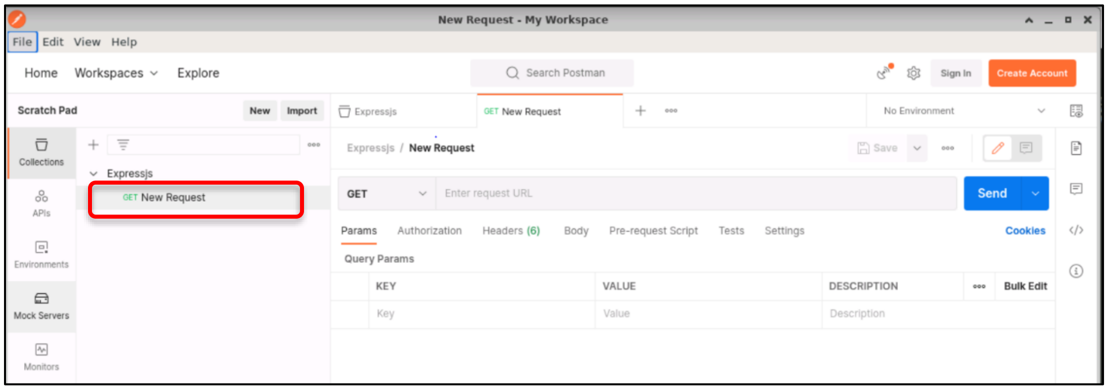


```
labuser@ubuntu2204:~$ sudo snap install postman
postman (v10/stable) 10.10.3 from Postman, Inc. (postman-inc✓) installed
labuser@ubuntu2204:~$
```

3.2 Navigate to **Application Finder** and open **Postman**



3.3 In the Postman workspace, create a collection for Express.js, run the request to check the responses of the functions, and subsequently add a new request to that collection



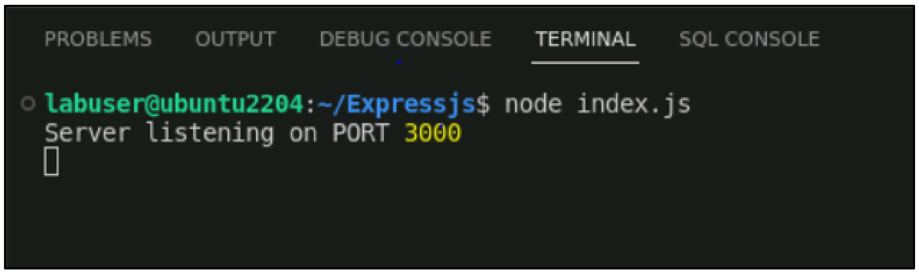
Note: For checking the response method, the Postman application is used.

Step 4: Handle GET and POST requests through Postman

4.1 Open the Expressjs folder in the VS code and write the below code in the **index.js** file:

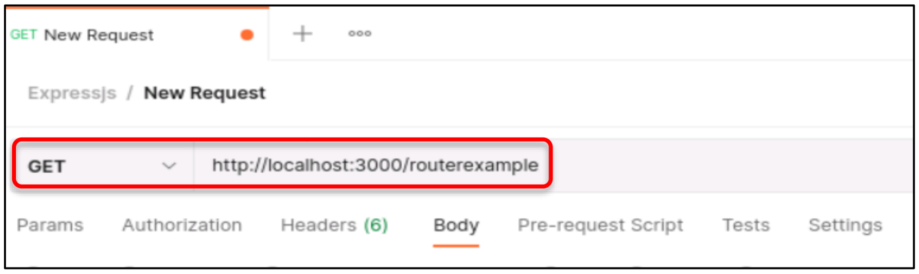
```
var express = require('express');
var app = express();
var PORT = 3000;

app.route('/routerexample)
.get((req, res, next) => {
   console.log("GET request called");
   res.send('GET request called');
})
.post((req, res, next) => {
   console.log("POST request called");
res.send('POST request called');
})

app.listen(PORT, function(err){
   if (err) console.log(err);
   console.log("Server listening on PORT", PORT);
});
```

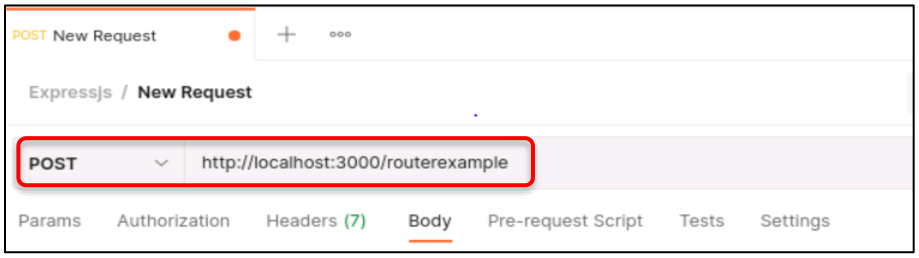
4.2 Run the **node index.js** command in the terminal

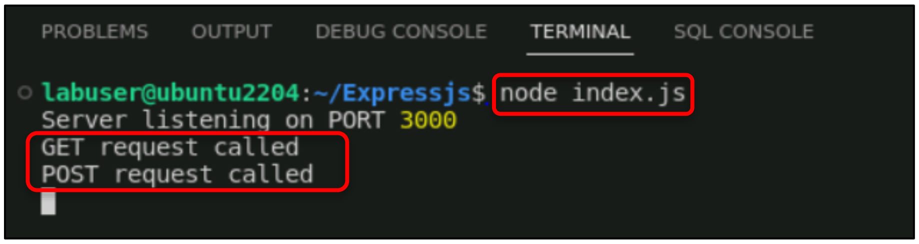


```
labuser@ubuntu2204:~/Expressjs$ node index.js
Server listening on PORT 3000
```

4.3 Open Postman, create a new request, execute GET and POST requests to **http://localhost:3000/**, and inspect the output in the VS Code terminal where the program is being executed







```
labuser@ubuntu2204:~/Expressjs$ node index.js
Server listening on PORT 3000
GET request called
POST request called
```

By following these steps, you have successfully demonstrated GET and POST request handling in an Express.js application to ensure functionality and response verification.