|  |  |
| --- | --- |
| Roll No. | 75 |
| Student Name | Prasad Kashid |
| Subject | Internet Programming |
| Date Of Experiment |  |
| Date Of Submission |  |

**SUBJECT: INTERNET PROGRAMMING EXPERIMENT NO. 9**

**AIM:** Implement Express Router

**OBJECTIVE:** To implement routing using express router.

**LAB OUTCOMES:** LO6

Construct front end applications using Node.js/Express

**THEORY:**

**What is express?**

Express is the most popular Node.js framework because it requires minimum setup to start an application or an API and is fast, and unopinionated at the same time. In other words, it does not enforces its own philosophy that a application or API should be built in a specific way, unlike Rails and Django. Its flexibility can be calculated by the number of npm modules available which makes it pluggable at the same time.

Express lets you build single page, multi-page, and hybrid web and mobile applications. Other common backend use is to provide an API for a client (whether web or mobile).

**What is routing?**

Routing is a method that refers to determining how an application responds to a client request to a particular path and a specific HTTP request method (GET, POST, and so on). There is a special routing method, app.all(), used to load middleware functions at a path for *all* HTTP request methods. For example, the following handler is executed for requests to the route “/secret” whether using GET, POST, PUT, DELETE, or any other HTTP request method supported in the http module.

In simple terms, routing is controlling which function gets invoked whenever the user navigates to a particular URL. In this context, URL refers to any path or route.

Routers are helpful in separating concerns such as different endpoints and keep relevant portions of the source code together. They help in building maintainable code. All routes are defined before the function call of app.listen(). In a typical Express application, app.listen() will be last function to execute.

**Defining Routing Methods** Routing methods are defined by :- *app.METHOD(PATH,CALLBACK)*

The above function tells the server, “If a user navigates to ***PATH***, then perform the following ***CALLBACK*** function and perform an ***HTTP METHOD*** request.”

Here, the most commonly used METHODs are:

**get** : To handle GET requests (i.e. to request/GET data from a specified resource).

*Syntax – app.get(path, callback)*

**post** : To send data to a server to create/update a resource.

*Syntax – app.post(path, callback)*

**put** : To send data to a server to create/update a resource.

*Syntax – app.put(path, callback)*

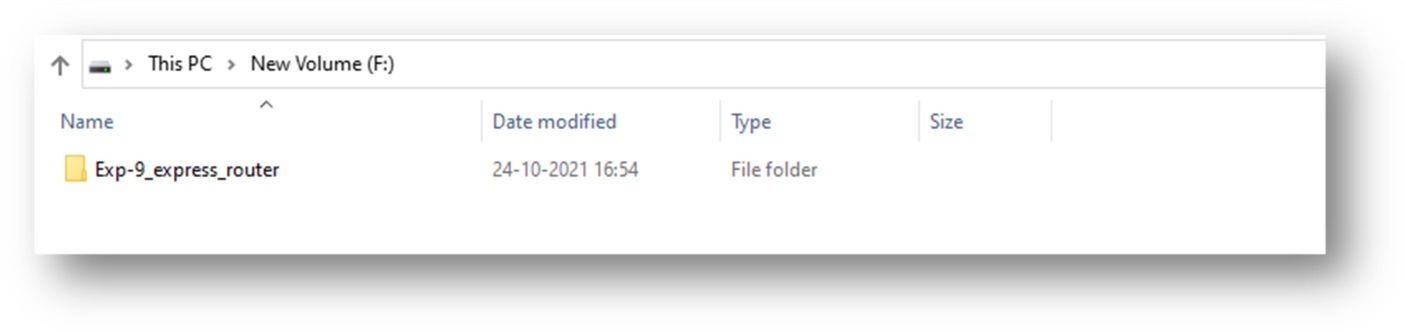
**NOTE:** The difference between POST and PUT requests is that the latter are idempotent. This means that PUT requests have no additional effect if they are called multiple times. In contrast, if you call a POST method more than once, your program will have side effects. Therefore, bear in mind that POST requests are not to be called more than once.

**delete** : For removal of a specified resource.

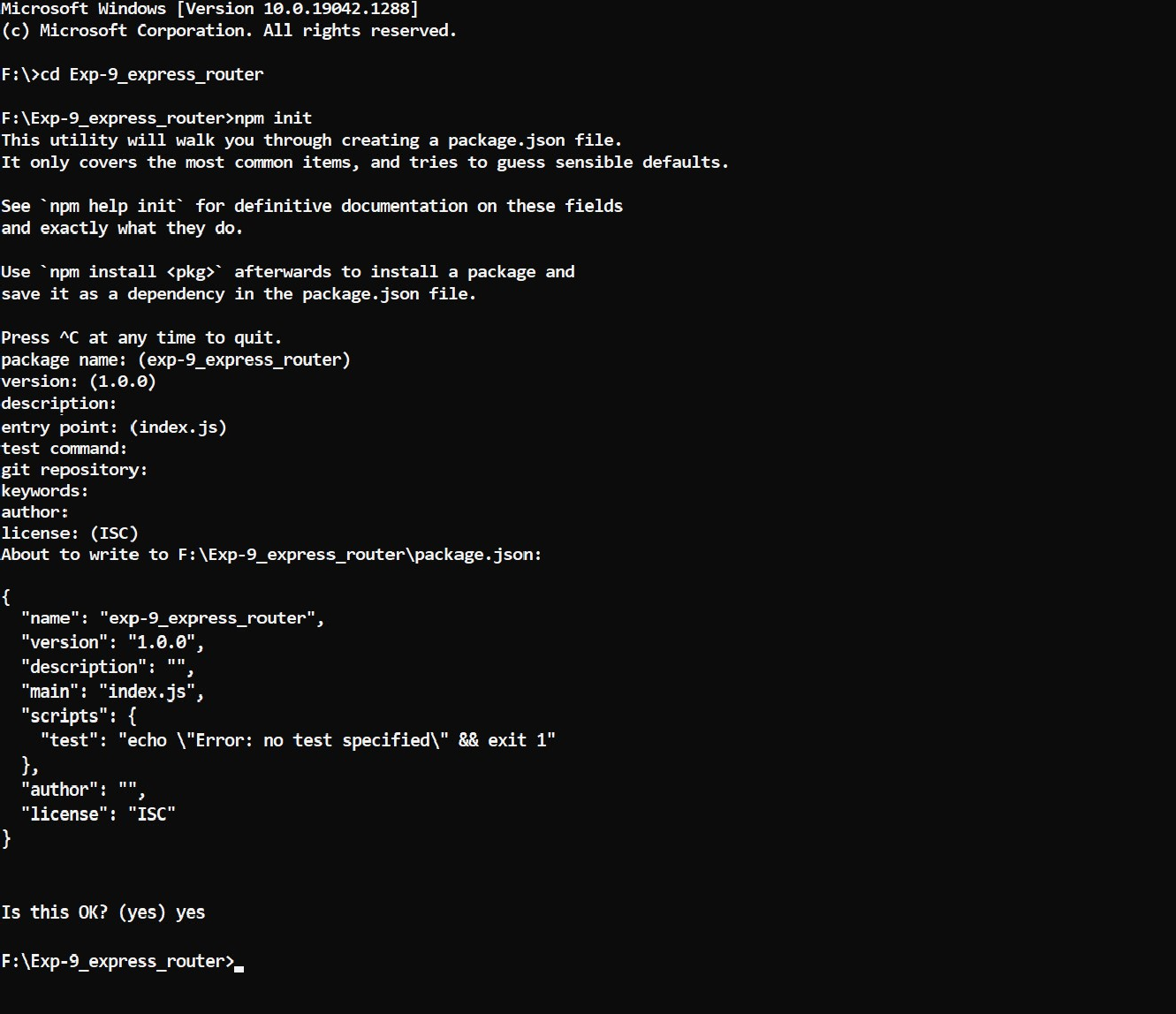
*Syntax – app.delete(path, callback)*

**IMPLEMENTATION:**

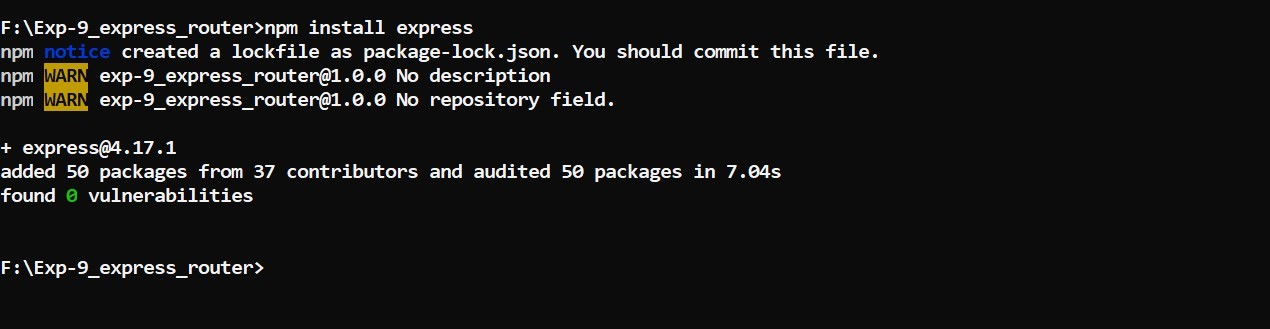
**Step 1:** Create a folder/directory



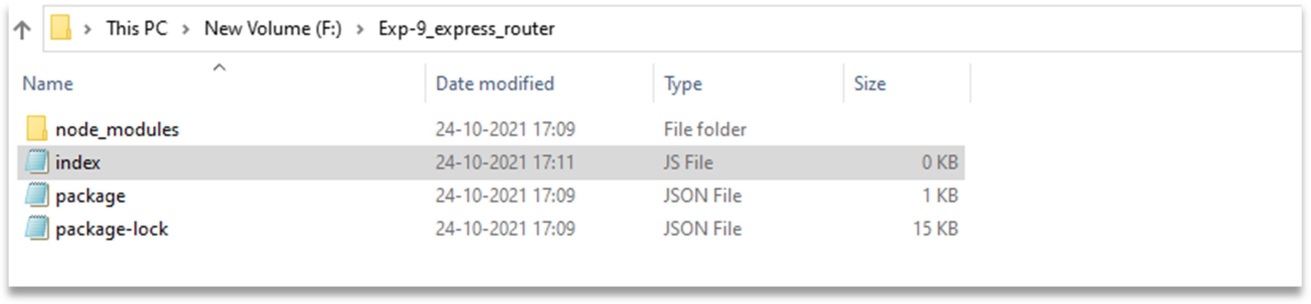
**Step 2:** Go to file location in command and run “npm init”. This will create a package.json file in the created directory.



**Step 3:** Go ahead and run “npm install express”. Now we have Express. Let's look at starting up our application and then we can handle our routing.



**Step 4:** To start up our application, we need to create index.js file in the created directory. All the programming will be done in this file.



**Step 5:** Open index.js file and start coding to implement express router.

**CODE:**

**index.js**  **app.get(path, handler)**

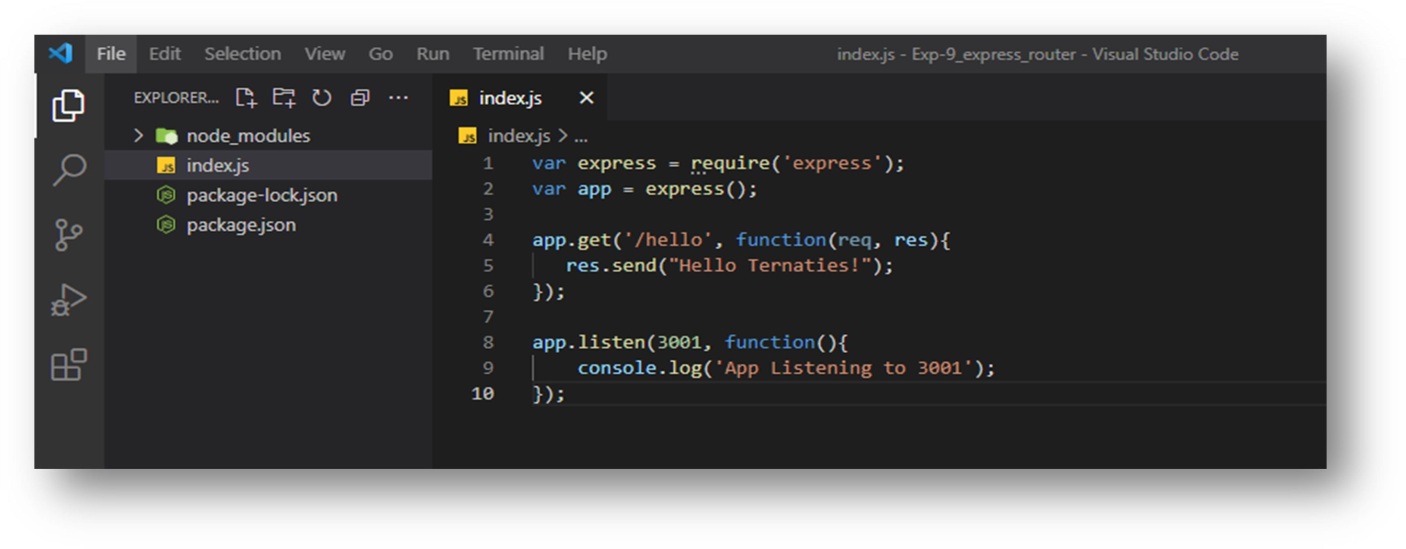
var express = require('express'); var app = express(); app.get('/hello', function(req, res){

res.send("Hello Ternaties!");

});

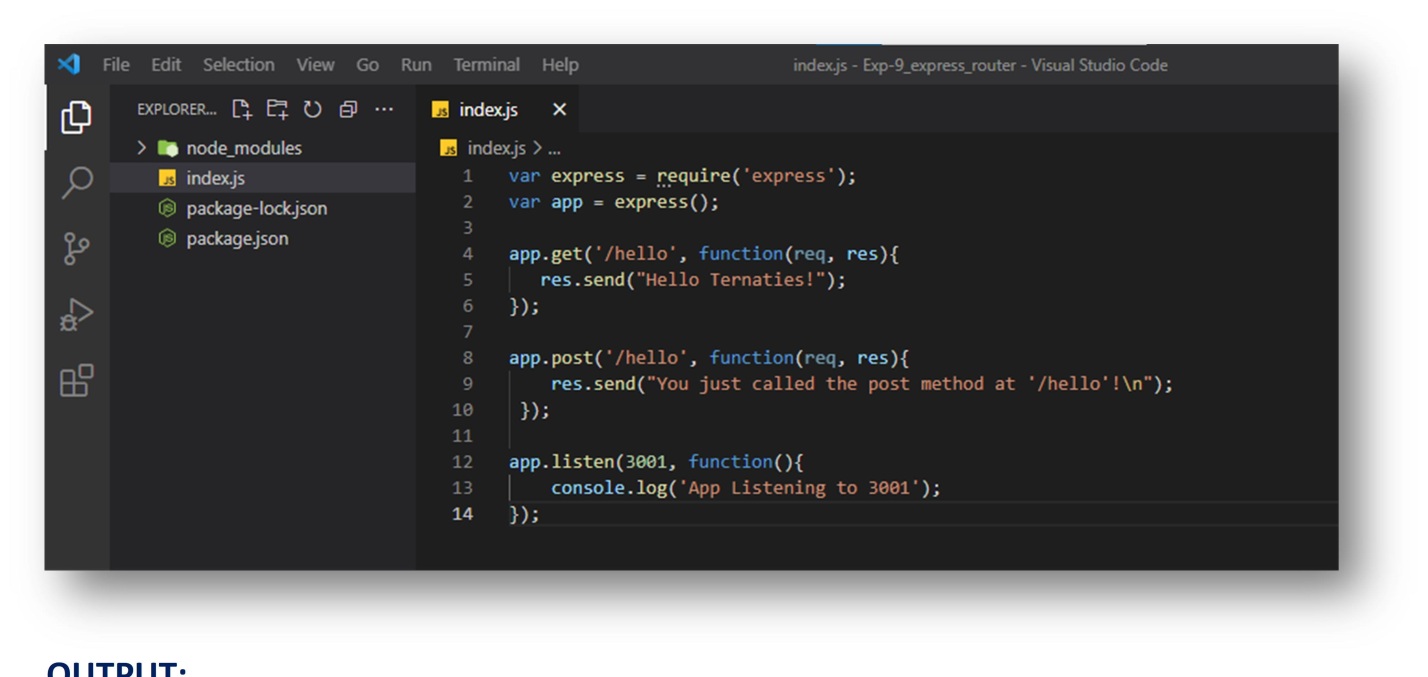
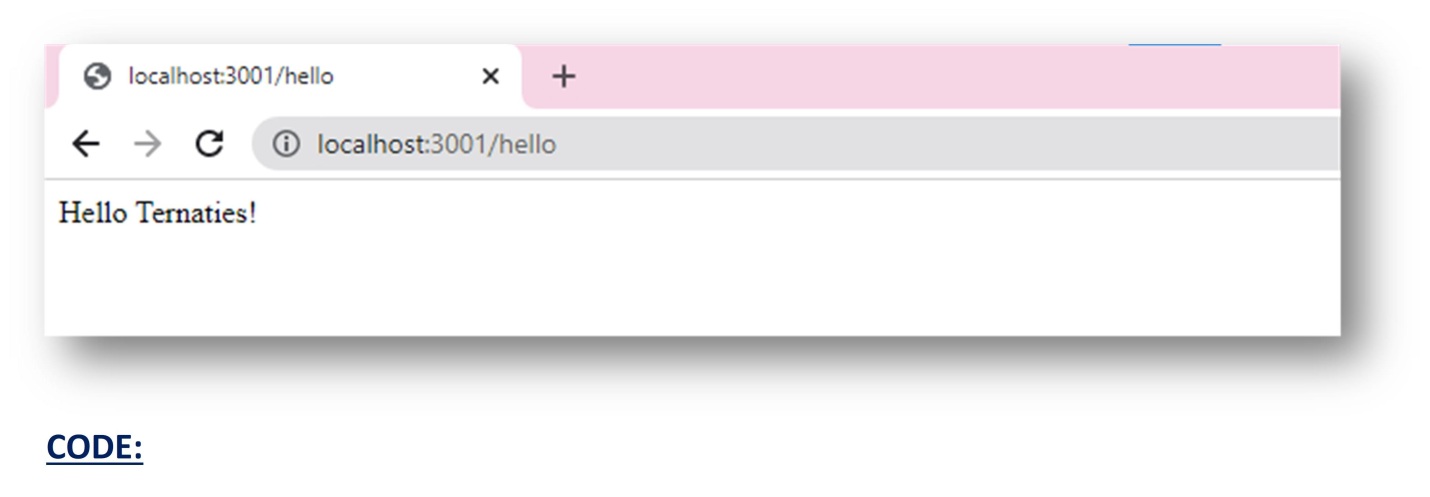
app.listen(3001, function(){ console.log('App Listening to 3001');

});



**OUTPUT:**





**CODE:**

**app.post(path, handler)**

var express = require('express'); var app = express();

app.get('/hello', function(req, res){ res.send("Hello Ternaties!");

});

app.post('/hello', function(req, res){

res.send("You just called the post method at '/hello'!\n");

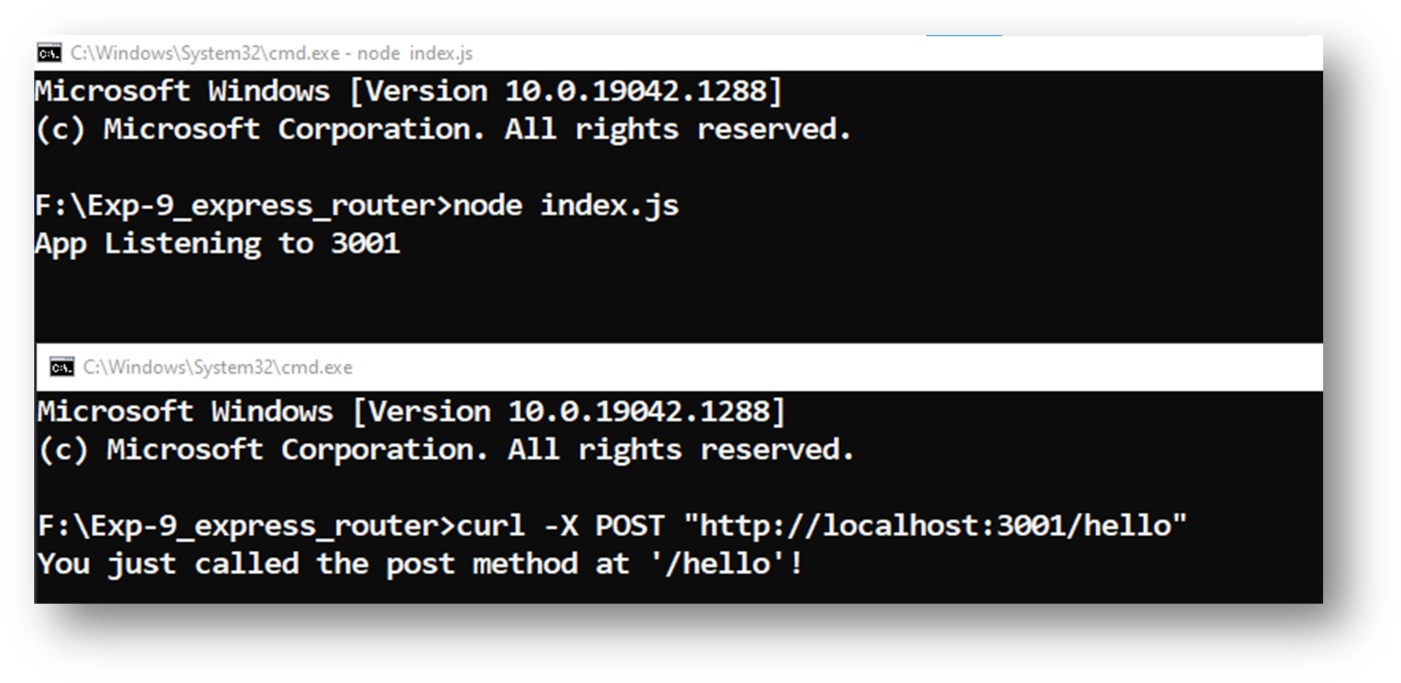
});

app.listen(3001, function(){ console.log('App Listening to 3001');

});

**OUTPUT:**

To test this request, open up your terminal and use curl to execute the following request –



**Step 6:** A routing path is a combination of a request method to define the endpoints at which requests can be made by a client. Let us define two more endpoints in our server based application.

**CODE:**

var express = require('express'); var app = express();

app.get('/hello', function(req, res){ res.send("Hello World!");

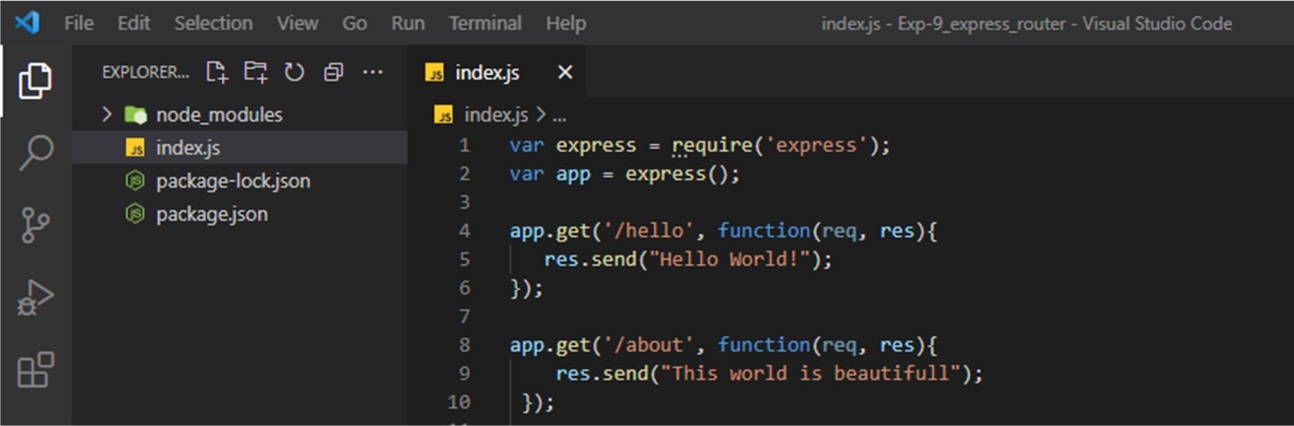
});

app.get('/about', function(req, res){ res.send("This world is beautifull");

});

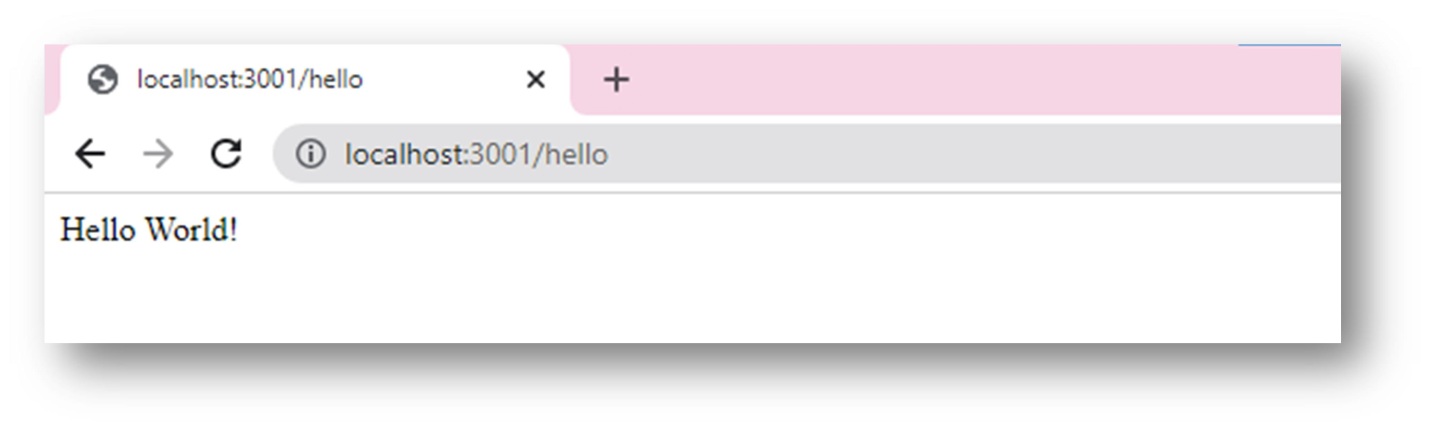
app.listen(3001, function(){ console.log('App Listening to 3001');

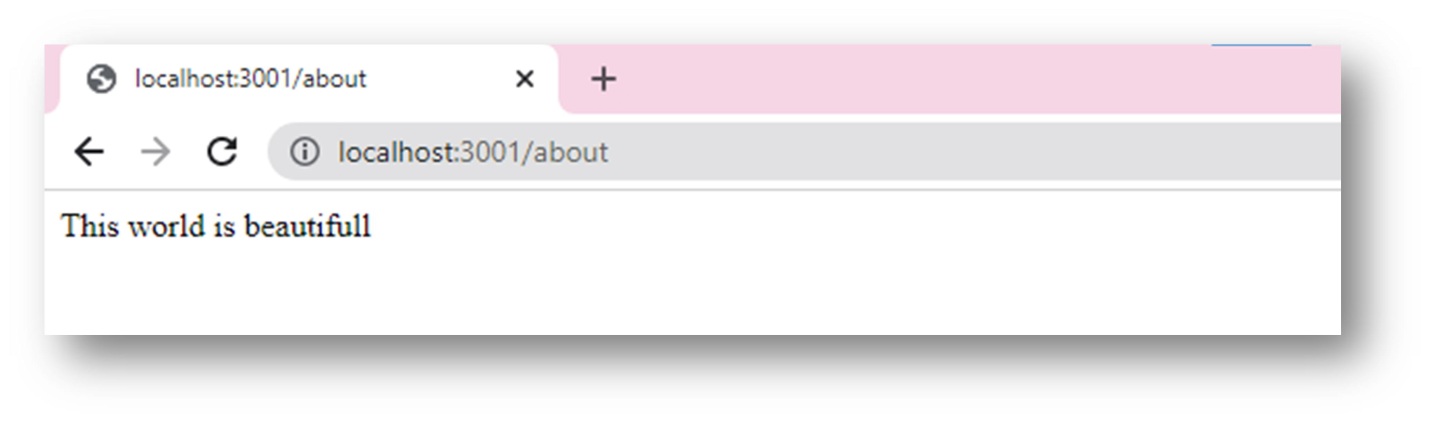
});





**OUTPUT:**

**localhost: 3001/hello**

**localhost: 3001/about**

**CONCLUSION:** In the above experiment we have studied about how to implement routing in express.js and different methods of routing.