Lab tutorial Week4

How to use Node JS Express to Create REST API

Prepared by Dr.Nadeem Qazi

CN5006

What to submit for this week's Lab Work

Save your work, write a reflective report, and upload the code to GitHub for your portfolio. Remember, you must complete your work during the lab session. If you do not attend, this week's lab will be penalized. Please note, your entire portfolio of lab work (Weeks 1 to 11) is worth 15 marks

---

In this exercise you will learn the working principle of the express framework REST API methods

After the completion of this Lab you will be able to do the following:

1. **Creating and express web application in node js**
2. **How to use the "get method" to create REST API URL with or without parameters**
3. **Use of body parser when you use the form to submit the data**
4. **How to read the parameter using req.param and req.paramsid**
5. **How to create JSON file and search it from server**
6. **How to read the JSON file using the express package fs.**

- **Step 1 Open Visual Studio code**
- **Step 2 : create a directory/folder Labsession or any other of your choice**
- **Step 4 : in your visual studio code switch to the folder you just created i.e Labsession**
- **Step 3: move into that directory ,write npm init command and follow the default entry ,it will create a package.json file**
- **Step 5: Install express package. type the following command in your Visual Studio code terminal: npm install express –save or npm install express -g**
- **Step 5: create index.js and start coding following into index.js**

Lets start it: before you write the code please note // is used to comment in JS and it is not treated as CODE

Exercise 1: write the code for the basic **express hello application** . NodeJS Core module used: "express"

```
var express=require("express")
var fs= require("fs")
var app=express()
// add middle ware function for body parsing

var bodyParser = require("body-parser");
app.use(bodyParser.urlencoded({ extended: true }));

app.get('/',function(req,res){
res.send("hello it is my first express application")
})
app.listen(5000,function(){console.log("server is runnin
g on port 5000")})
```

How to Run:

Save the file by clicking Save in the File menu option in your Visual Studio code. Then in t h e terminal window write the following:

node index.js

You will see a console message saying server is running on port 5000.
Now open google chrome or any other browser and in the location bar ofthe browser type this localhost:5000
This is the IP address of the application that we have just created and 5000 is the port number. On the browser window, you will see the following message

Hello it is my first express application

Exercise 2:

Now return to Visual Studio code again, press Ctrl C to terminate the running application, **and add the following code in index.js**. Add two more URLs in the index.js using following code:

app.get('/about',function(req,res)

{ res.send("This is basic express application ")

```
})
app.get('/users/:userId/books/:bookId', function (req, res) {
res.send(req.params)
})
```

Once again save all the files. To run this application once again go to the terminal window and write node index.js and you will see the server message again.

Now open the browser and test following urls

1. **Localhost:5000**
2. **Localhost:5000/about**
3. **Localhost:5000/users/33/books/123**

Report back what you observed, also try different userid and bookid such as

1. Localhost:5000/users/334/books/143 Note it can be any number.
4. Also try this Localhost:5000/user/33/books/123 : Observe and report what you find on the browser

Now once again switch back to the Visual Studio code and press ctrl C to terminate the runningapplication. write the following code in Student.json for the exercise 3.

Exercise 3

Step 1: Create a file name Student.json and add the following code to it:

```
{
    "Student1" : {
    "name" : "Jonhthon",
    "Age" : "33",
    "Qualification" : "BSC",
    "Email": "std123@gm.com", "id": 1
    },
    "Student2" :
    { "name" : "David",
    "Age" : "23",
    "Qualification" : "HNC", "Email": "Abc@gm.com", "id"
: 2
    },
    "Student3" :
    { "name" : "Emily",
    "Age" : "25",
```

```
    "Qualification" : "A-
level", "Email": "email@gm.com", "id": 3
    }
 }
```

Step 2: Read this JSON file in your web application appending the following code in your index.js

Note in this code you will be using __dirname. The __dirname in a node script returns the path of the folder where the current JavaScript file resides. __dirname is used to get the directory name of the currently executing file.

```
app.get('/GetStudents',function (req,res)
 { studentdata={}
  fs.readFile(__dirname + "/" + "Student.json", 'utf8',
function (err, data) { console.log( data );
  res.json({ 'status':true, 'Status_Code':200,
  'requested at': req.localtime, 'requrl':req.url,
  'request Method':req.method, 'studentdata':JSON.parse(
data)});
});
})
```

Save all the files and Run the application again. use node index.js

Open the browser window and observe and report what happens when you type the following

Localhost:5000/GetStudents

Report the output of your browser window what did you see when you run this URL?

Step 3 Searching the JSON File :Press ctrl C and add more code in the index.js

```
app.get('/GetStudentid/:id',(req,res)=>{
    studentdata={}
    fs.readFile(__dirname + "/" + "Student.json", 'utf8'
, function (err, data) {

    var students= JSON.parse(data)
    var student=students["Student"+req.params.id]
    console.log("student",student)
    if (student)

    res.json(student)
    else
    res.json({ 'status':true, 'Status_Code':200,
    'requested at': req.localtime, 'requrl':req.url,
    'request Method':req.method, 'studentdata':JSON.pars
e(data)});
    });


    })
```

Run and Report

Save all the files and write node index.js and after you received server is running message open the browser and type following URL

- Localhost:5000/ GetStudentid/1 and report what you observed
- Localhost:5000/ GetStudentid/2 and report what you observed
- Localhost:5000/ GetStudentid/3 and report what you observed
- Localhost:5000/ GetStudentid/4 and report what you observed

Report what did you observe for following GetStudent/1, GetStudent/2 ,GetStudent/4 one by one and explain your output

Once again press ctrl C to terminate the running application and follow these steps in visual studio code.

Exercise 3

Use of Post Method

Create a HTML file **name it StudentInfo.html** and write following code

E**ither you can type it or you can download this file from week 4 and copy it into same folder where you wrote todays code**.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta charset="utf-8" />
    <title>Student Enrollment Form</title>
</head>
```

```
<body bgcolor= beige>
    <form action="/submit-data" method="post" >
<p><Strong> Student Details </p>
        First Name: <input name="firstName" type="text" /> <br/><br/>
        Last Name : <input name="lastName" type="text" /> <br/><br/>
        Email:         <input type="email" id="emailid"
name="email"/><br/> <br/>
        Age :     <input name="myAge" type="text" /> <br/>

 <p>Please select your gender:</p>

  <input type="radio" id="male" name="gender" value="male">
  <label for="male">Male</label><br>
  <input type="radio" id="female" name="gender" value="female">
  <label for="female">Female</label><br>
  <input type="radio" id="other" name="gender" value="other">
  <label for="other">Other</label>

<p> Qualifications </p>
<input type="checkbox" name="Qual" value="GCSE"> GCSE <br/>
  <input type="checkbox" name="Qual" value="A-level"> A- level <br>
  <input type="checkbox" name="Qual" value="Higher National Certificate">
Higher National Certificate/Level 4 <br/>
  <input type="checkbox" name="Qual" value="HND"> Foundation
Degree/HND/DipHE/Level 5 <br/>
  <input type="checkbox" name="Qual" value="B.Sc"> Bachelor Degree/Graduate
diploma or
Certificate/Level 6 <br/>

  <input type="checkbox" name="Qual" value="Master Degree"> Master
Degree/PGCE/Level7 <br/>
  <input type="checkbox" name="Qual" value="PhD"> PhD/Level8 <br/>
 <input type="submit" />
    </form>
</body>
</html>
```

Step 2 append following method in index.js

```
app.get('/studentinfo',function(req,res)

{
res.sendFile('StudentInfo.html', { root:    __dirname });

})
```

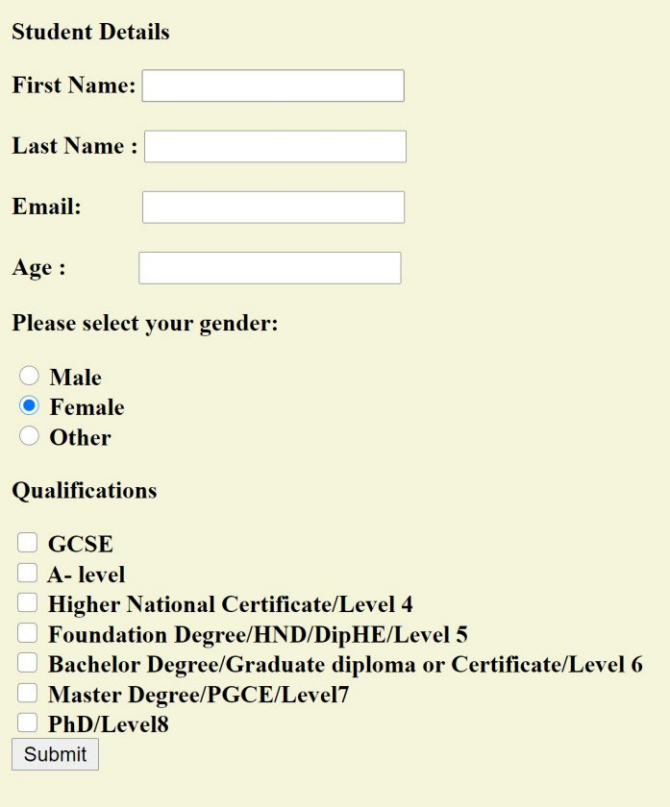
```
app.post('/submit-data', function (req, res) {
var name = req.body.firstName + ' ' + req.body.lastName+
 ' ';

var Age= req.body.myAge+ ' Gender: ' + req.body.gender+'
 '
Qual= ' Qualification'+ req.body.Qual
console.log(req.body.Qual)
res.send({
status: true,
message: 'form Details', data: {
name: name, age:Age, Qualification:Qual,
}
});
});
```

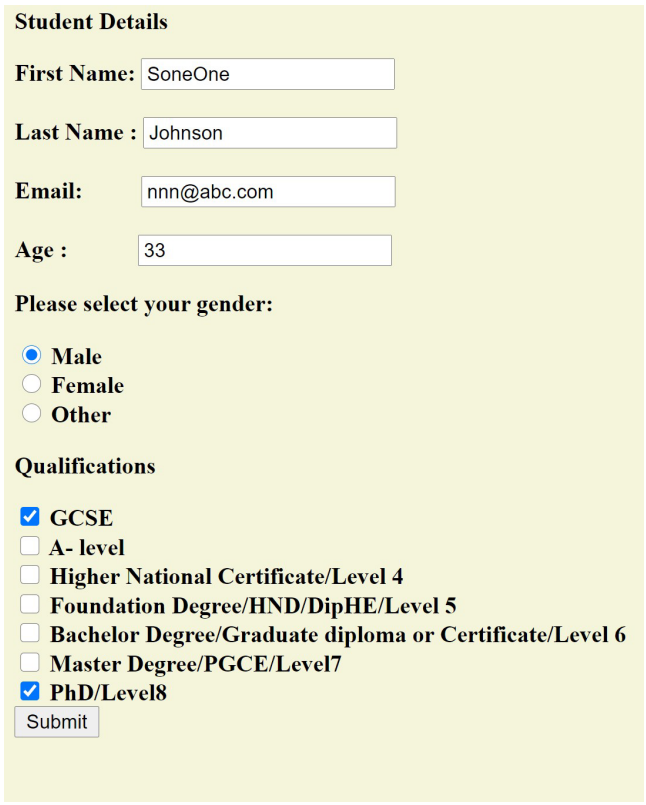
Save all the files and write node index.js to start the application then open the browser window and type

Localhost:5000/ studentinfo

You will observe the following form



Fill this form with appropriate dummy data like this



Submit this form clicking the submit button

You may see following output based on what you entered.:


```
{"status":true,"message":"form Details","data":{"name":"SoneOne Johnson
","age":"33 Gender: male ","Qualification":" QualificationGCSE,PhD"}}
```


What to submit for this week Lab Work

Save your work, write a reflective report, and upload the code to GitHub for your portfolio. Remember, you must complete your work during the lab session. If you do not attend, this week's lab will be penalized.

Please note, your entire portfolio of lab work (Weeks 1 to 11) is worth 15 marks.