Getting Started with Express.js Lab

How to set up Express that works within Node. JS and its basic functions.

Express is the most widely adopted web framework for Node.js. Node.js itself does not provide direct support for many other web development tasks. For instance, if you aim to incorporate specific handling for different HTTP verbs like GET, POST, DELETE, etc., handle requests individually based on distinct URL paths (known as "routes"), serve static files, or utilize templates to dynamically generate responses, relying solely on Node.js won't be sufficient. In such cases, you have two options: either write the necessary code yourself or leverage the capabilities of a web framework.

First application using Express

Start by creating a directory for the application called express-app, and then navigate to this directory using the following commands on the terminal.

mkdir express-app

cd express-app

Create a **package.json** application configuration using following command

npm init -y

Express is a Node module, and you can install it using **npm.**

Run the following command on terminal to install express. --save flag directs npm to save it as a dependency in packages.json file.

npm install express --save

Importing Modules

A module is a JavaScript library/file that can be imported into other code using Node's **require()** function. Express itself is a module, as are the middleware and database libraries that we use in our Express applications. For example,

```
const express = require("express");

const app = express();
```

Using Visual Studio Code editor, create a file **express-app.js** and add the following code to it.

```
const express = require("express");
const app = express();
const port = 8080;
app.get("/", function (req, res) {
    res.send("Welcome to Express!");
});
app.listen(port, function() {
    console.log(`App listening on port ${port}!`);
    console.log(`Hello World!`);
});
```

Run the application by using the command **node express-app.js** in terminal and place screenshot below:

Navigate to http://localhost:8080/. using any browser to start the server, and place screenshot below:

Express Application Generator

An Express application can be generated using a utility called Express Application Generator. It quickly creates a skeleton application generator.

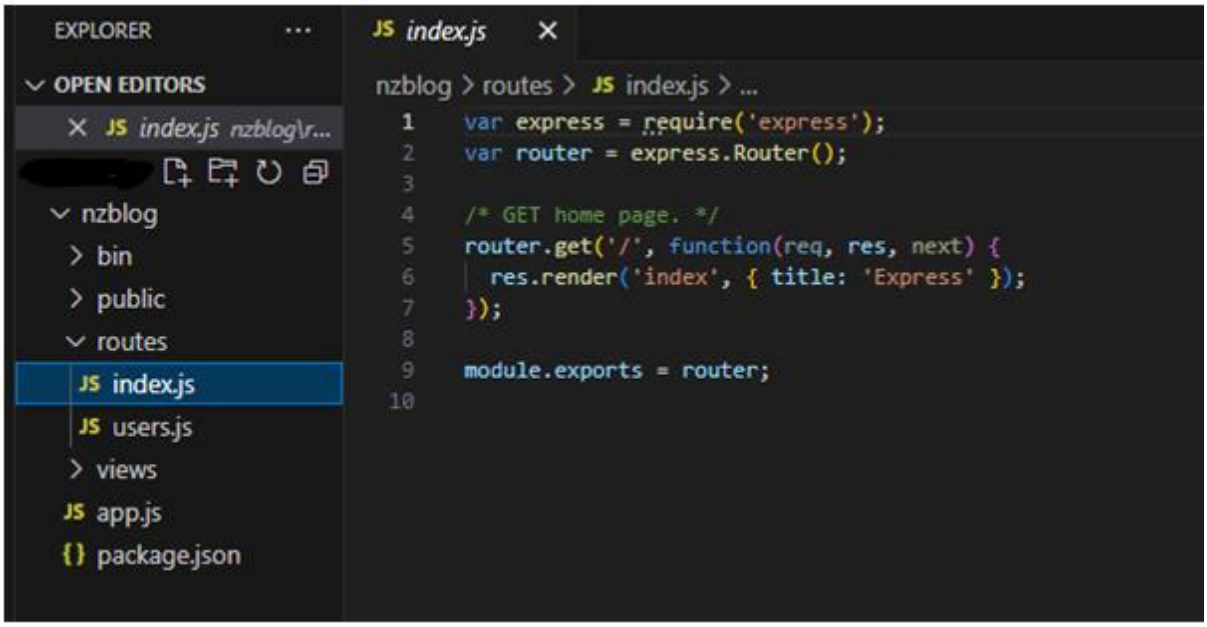
To use the application generator, input the following commands

npm install express-generator -g

To create the app, navigate to any directory to create the app and run the following command in the terminal.

npx express-generator nzblog

Open the newly created folder nzblog with VS code, and you will see the following files and directory structure.



Navigate to the nzblog directory and run the following command to install all dependencies

npm install

Run the following command to start the application. We do not have to profile file name

npm start

Navigate to http://localhost:3000 using any browser to start the server and place screenshot below:

Add New Routes

Add new routes to handle different pages and learn routing in Express.

Create a New Route for Contact Page

Create route as contact.js

```
var express = require('express');
var router = express.Router();

router.get('/', function(req, res, next) {
   res.send('Please contact us at ITSupport@whitecliffe.com');
  });

module.exports = router ;
```

Update app.js to Use the New Route

Add the following line to app.js

```
var contactRouter = require('./routes/contact');
app.use('/contact', contactRouter);
```

Test the New Route

Navigate to http://localhost:3000/contact in browser to start the server and place screenshot below:

Task: Create a simple form that submits data to the server.