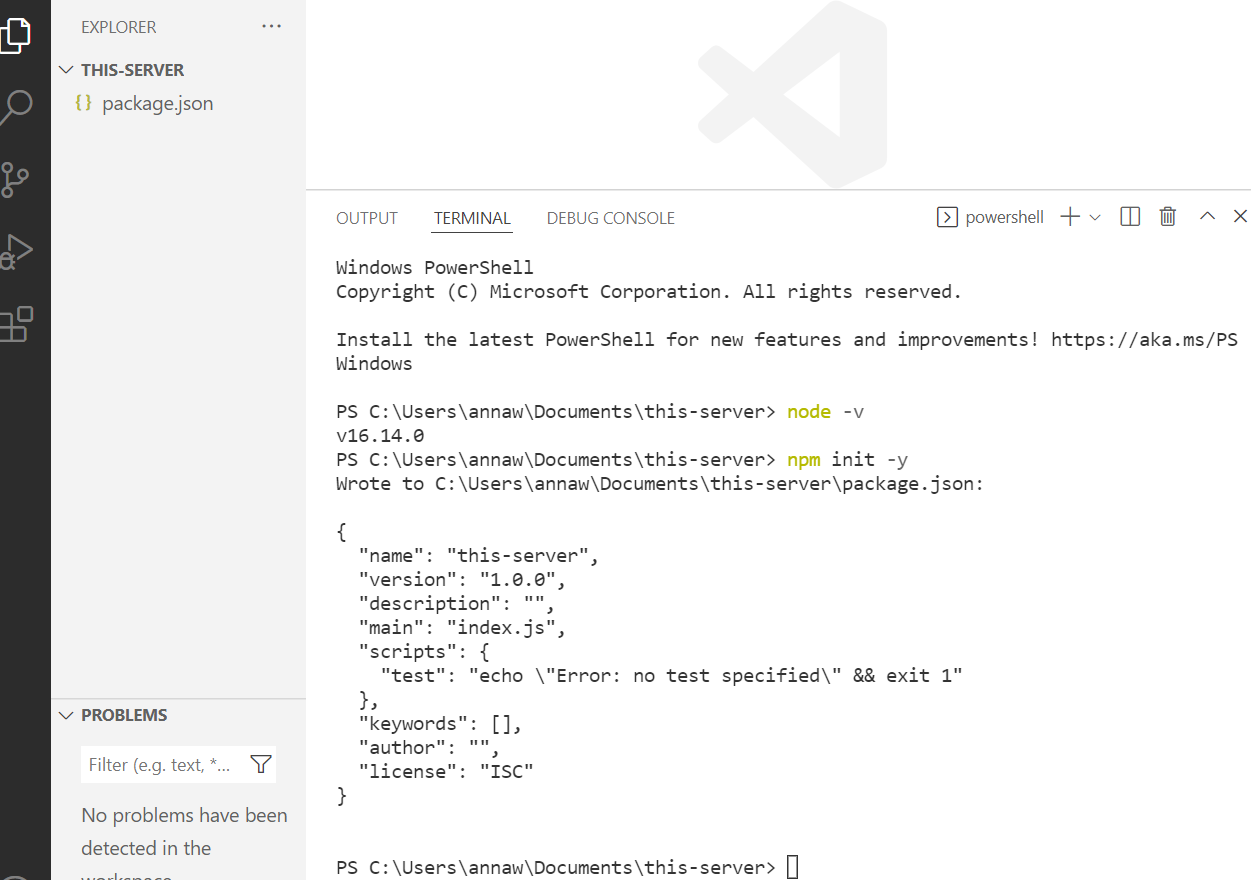
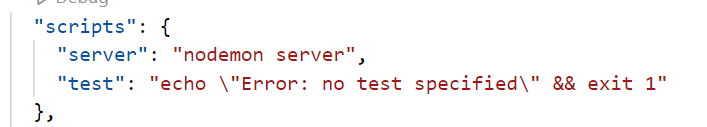
This will be a series of lessons that will yield a backend server using NodeJS, Express, SQL-Server.

1. Ensure that NodeJS is installed on your computer. In the VS CODE terminal, type in   
    node -v if NodeJS is installed you will see a ‘version number’
2. Create a folder called ‘this-server’ (no capital letters)
3. Open Visual Studio Code, then in VS-CODE, open ‘this-server’. This will be the name of the project and will appear as “THIS-SERVER” in visual studio code at the root level.
4. VS CODE Terminal: type in npm init -y to initialize the environment. This will create a file called ‘package.json’ below the root level.  
     
     
   
5. In the package.json file, enter “server” : “nodemon server”, key-value pair in ‘scripts’ Notice that you must enter a comma “,” after the closing quotation mark. SAVE the file.

With VS-



1. Open the VS-CODE Terminal to install the ExpressJS package, CORS package, Nodemon package, body-parser package, and the Microsoft SQL-server Interface Package, With the terminal enter npm install express cors mssql body-parser nodemon

Express: is the NodeJS framework,

CORS: allows the server to communicate with external

Client Front End

Nodemon: NODE-MONITOR allows the server to run continuously

Mssql: allows the server to connect to Microsoft MS SQL-Server.

Body-parser: is the tool to allow the server to parse json data.

1. Create a file called “server.js”
2. In ‘server.js’, Enter the following code and SAVE it. NOTICE: USE BACK-TICS (“` “ to the left of the number 1 on your keyboard). In app.listen DO NOT USE regular “ ‘ “.  
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
   9) Run the Server in the Terminal. Enter nodemon server

const express = require('express'); // import express

const cors = require('cors'); // import cors

const API\_PORT = process.env.PORT || 5000; // Run the server on Port 5000

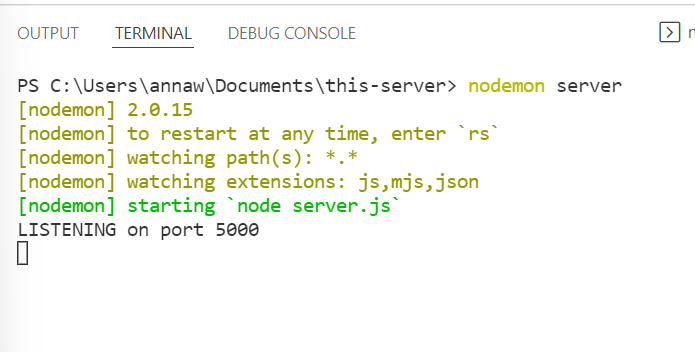
const app = express(); // activate express as ‘app’

app.use(cors()); // activate CORS

// activate the server to listen on port 5000

app.listen(API\_PORT, ()=> { console.log(`LISTENING on port ${API\_PORT}`) });

A successful run will produce the following “LISTENING on port 5000” message  
  
   
  
  
  
  
  
  
  
  
  
  
  
NOTE: to stop nodemon - in the terminal, enter “control c” (ctrl c).  
10) Add an ‘api’ endpoint to server.js (in express the endpoint called an ‘api route’). The api endpoint that we want to create is /user



const express = require('express');   // import express

const cors = require('cors');      // import   cors

const API\_PORT = process.env.PORT || 5000;  // Run the server on Port 5000

const app = express();                // activate express as ‘app’

app.use(cors());  // activate   CORS

// add an api route

app.get('/user', (req, res)=> {

    console.log(" this is your first USER api");

    res.send('USER RESPONSE sends the message to the external BROWSER');

})

app.listen(API\_PORT, ()=> { console.log(`LISTENING on port ${API\_PORT}`) });

**TESTING THE API:**

1. In VS-CODE Terminal: enter nodemon server

TEST 1: Open a new Chrome Browser window

In the url window type: <http://localhost:5000/user>

Note: Do NOT enter https use only http

*VS CODE terminal* will display

[nodemon] restarting due to changes...

[nodemon] starting `node server.js`

LISTENING on port 5000

this is your first USER api

The *Chrome* browser will display the message



TEST 2: Download POSTMAN and test using <http://localhost:5000/user>

