

## Lesson 04 Demo 02

### Creating a Node Server

**Objective:** To establish a NodeJS server with Express to communicate with the model layer data

**Tools required:** VS Code and NodeJS

**Prerequisites:** Basic understanding on JavaScript

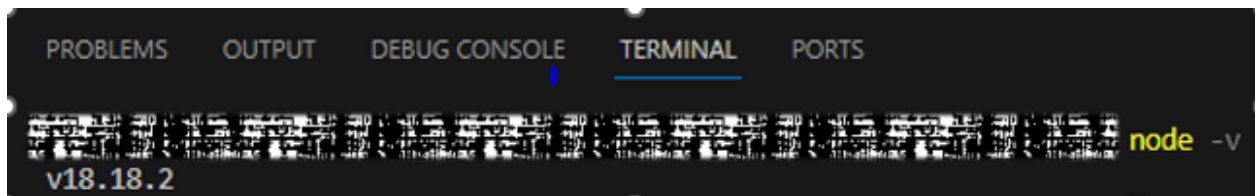
Steps to be followed:

1. Check NodeJS availability
2. Set up the server

#### Step 1: Check NodeJS availability

- 1.1 Run the command given below to check if NodeJS is installed in the system. However, in the lab environment, NodeJS is already installed:

```
node -v
```



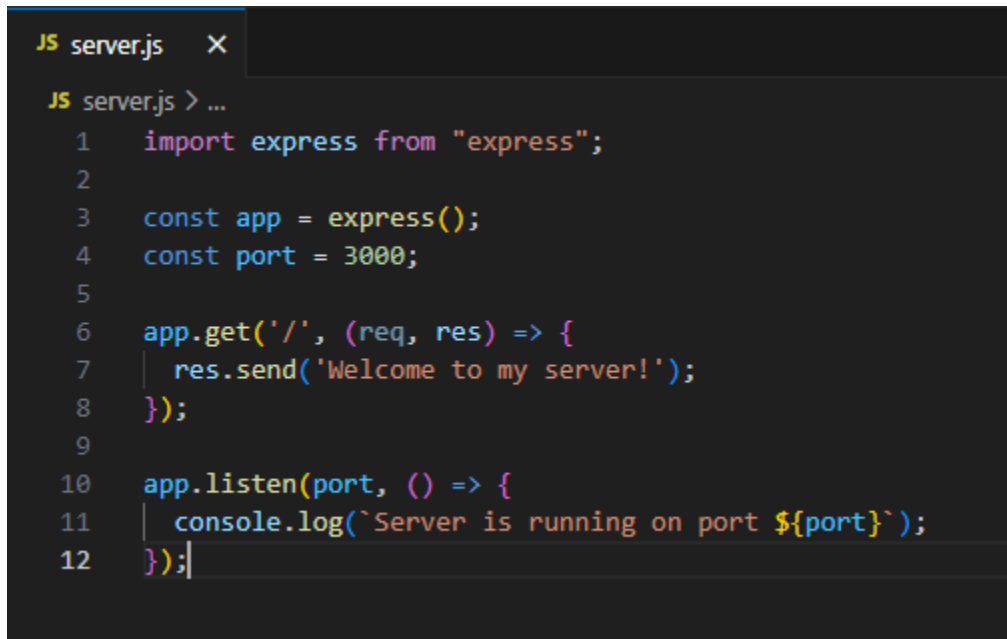
#### Step 2: Set up the server

- 2.1 Run the command given below to check if NodeJS is installed in the system. However, in the lab environment, NodeJS is already installed:
- 2.2 Create a new project directory with the name as **NodeJSExamples** and navigate to the same using the command given below to initialize NodeJS:

```
mkdir NodeJSExamples
cd NodeJSExamples
node init -y
npm install express
```

2.3 Create a JavaScript file with the name **server.mjs** and the code given below:

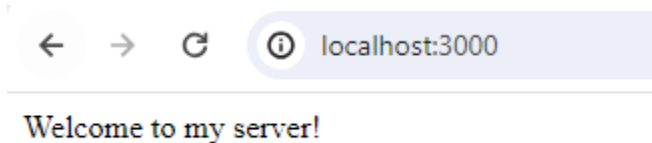
```
import express from "express";
const app = express();
const port = 3000;
app.get('/', (req, res) => {
  res.send('Welcome to my server!');
});
app.listen(port, () => {
  console.log(`Server is running on port ${port}`);
});
```

A screenshot of a code editor window titled 'JS server.js'. The editor shows the following JavaScript code:

```
1 import express from "express";
2
3 const app = express();
4 const port = 3000;
5
6 app.get('/', (req, res) => {
7   res.send('Welcome to my server!');
8 });
9
10 app.listen(port, () => {
11   console.log(`Server is running on port ${port}`);
12 });
```

2.4 Run the command given below to start the server. Once the server is running, open **localhost:3000** in the browser:

**node server.js**



By following these steps, you have successfully established a NodeJS server with Express to communicate with the model layer data.