

USER GUIDE

Setting up a Node.js Express server using the VS Code editor is a relatively straightforward process. Here's a step-by-step guide to help you get started:

****Step 1: Install Node.js****

Before setting up the Express server, you need to have Node.js installed on your machine. You can download the Node.js installer from the official Node.js website (<https://nodejs.org>) and follow the installation instructions for your operating system.

****Step 2: Create a New Project****

Open VS Code and create a new folder for your Express server project. You can do this by clicking on the "Explorer" icon on the left sidebar, right-clicking on a suitable location, and selecting "New Folder." Give your folder a name.

****Step 3: Initialize the Project****

Next, open a new terminal in VS Code by clicking on "View" in the top menu and selecting "Terminal" or using the shortcut `Ctrl+``. In the terminal, navigate to the project folder you created in the previous step using the `cd` command. For example:

```
```bash
cd path/to/your/project/folder
```
```

Once you're inside the project folder, run the following command to initialize a new Node.js project:

```
```bash
npm init -y
```
```

This command creates a `package.json` file, which will keep track of your project dependencies.

****Step 4: Install Express****

To install Express, run the following command in the terminal:

```
```bash
npm install express
```
```

This command downloads and installs the Express package and its dependencies into your project.

****Step 5: Create the Server File****

Create a new file in your project folder, e.g., `server.js`, which will serve as the entry point for your Express server. Open the file in VS Code.

****Step 6: Set Up the Server****

In the `server.js` file, require Express and create an instance of the Express application. Add the following code:

```
````javascript
const express = require('express');
const app = express();

// Define your routes and middleware here

// Start the server
const port = 3000;
app.listen(port, () => {
 console.log(`Server is running on port ${port}`);
});
````
```

You can modify the `port` variable to use a different port number if needed.

****Step 7: Define Routes and Middleware****

Now you can define your routes and middleware inside the `server.js` file. For example, to create a basic route that responds with "Hello, World!" when you visit the root URL, add the following code before the server start code:

```
````javascript
app.get('/', (req, res) => {
 res.send('Hello, World!');
});
````
```

You can add more routes and middleware as needed.

****Step 8: Start the Server****

To start your Express server, go to the terminal in VS Code and run the following command:

```
````bash
node server.js
````
```

You should see the message "Server is running on port 3000" in the terminal, indicating that your server is up and running.

That's it! You have successfully set up a basic Node.js Express server using the VS Code editor. You can now visit `http://localhost:3000` in your web browser to see the "Hello, World!" response. Feel free to expand on this setup by adding more routes, middleware, and functionality to meet your requirements.