

Day 3:

### What is Express?

Express.js is a **small and simple framework for Node.js** that helps you build websites and APIs easily.

In very easy words:

- ◆ Node.js lets you run JavaScript on the server.
- ◆ Express makes it easier to create a server and handle requests.

Think like this:

- **Node.js = Engine**
  - **Express = Tool that makes work easy**
- 

### Why do we use Express?

Without Express, creating a server in Node.js is a little complicated.

With Express, it becomes clean and simple.

Example:

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

```
const app = express();
```

```
app.get("/", (req, res) => {
```

```
  res.send("Hello World");
```

```
});
```

```
app.listen(3000);
```

### Real Life Example

Imagine a restaurant:

- Customer = Client
- Kitchen = Server
- Waiter = Express

Express works like a **waiter** —

It takes the request from the client and gives the response from the server.

## Why Express is Popular?

- Easy to learn
- Simple syntax
- Fast
- Used for APIs and backend development
- Very popular in MERN stack

## Full Proper Steps (if starting new project)

1. Create folder

```
mkdir myproject
```

2. Go inside folder

```
cd myproject
```

3. Initialize project

```
npm init -y
```

4. Install Express

```
npm install express
```

Result: Express installed.

You will see:

- node\_modules folder created
- package.json updated

The screenshot shows a terminal window with the following command history:

```
PS C:\Users\vanda\Desktop\Backend2\myproject> npm init -y
{
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}

● PS C:\Users\vanda\Desktop\Backend2\myproject> npm i express
added 65 packages, and audited 66 packages in 6s
22 packages are looking for funding
  run 'npm fund' for details
found 0 vulnerabilities
○ PS C:\Users\vanda\Desktop\Backend2\myproject>
```

The terminal is running on a Windows system (powershell) and shows the creation of a package.json file with basic configuration and the installation of the express package. The package.json content is also displayed above the command history.

## How to create your first Express server?

### Step 1: Create a file

Inside your project folder, create a file:

server.js

---

### Step 2: Add this code inside server.js

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

app.get("/", (req, res) => {
  res.send("Hello Express ");
});

app.listen(3000, () => {
  console.log("Server is running on port 3000");
});
```

---

### Step 3: Run the server

In terminal, type:

node server.js

If everything is correct, you'll see:

Server is running on port 3000

---

### Step 4: Open in browser

Go to:

<http://localhost:3000>

You'll see:

Hello Express

You just built your first backend server.

---

The screenshot shows the Visual Studio Code interface. The left sidebar displays a project structure for 'MYPROJECT' containing 'node\_modules', 'package-lock.json', 'package.json', and 'server.js'. The 'server.js' file is selected and open in the main editor area. The code in 'server.js' is:

```
1 const express = require("express");
2 const app = express();
3
4 app.get("/", (req, res) => {
5   res.send("Hello Express ");
6 });
7
8 app.listen(5000, () => {
9   console.log("Server is running on port 5000");
10 });
11
```

Below the editor, the 'TERMINAL' tab is active, showing command-line logs from PowerShell (PS). It shows two entries:

- PS C:\Users\vanda\Desktop\Backend2\myproject> node server.js  
Server is running on port 3000
- PS C:\Users\vanda\Desktop\Backend2\myproject> node server.js  
Server is running on port 5000

#### Step 4: Open in browser

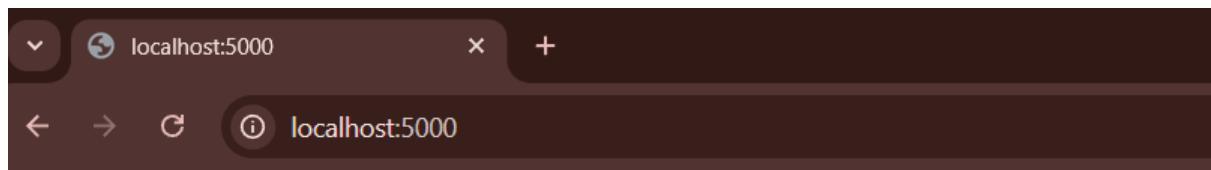
Go to:

<http://localhost:5000>

You'll see:

Hello Express

You just built your first backend server.



Hello Express

### What just happened?

- `express()` → creates server
- `app.get()` → handles request
- `res.send()` → sends response
- `app.listen()` → starts server