

Episode - 3

Creating our Express Server

1) what is express?

- Express is a web application framework build on top of Node.js.
- It provide features for creating web server API , Routing , Middleware & so on .
- It is ideal for creating REST API's, allowing developers to define routes that correspond to different HTTP requests methods.

Creating our Express server:

Step 1: Initialize the project

npm init

- It creates a "package.json" file

Step 2: Creating a src folder & new file in it

Step 3: Installing Express js

npm i express

- It creates a "node-modules" folder & "package-lock.json" file.

Step 4: Creating server

const express = require("express") → Importing express

const app = express() → new instance of express is created

// Defining routes

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

| → Request handler
function.

3) res.send("Hello from the server")

↓
Sending response back to client

"Creating/Starting Server:

app.listen(portNumber, () => console.log("Started server"))

→ app.listen() starts the server at the specified port number and listens for incoming requests.

2) what is Node-modules folder?

→ This folder contains all the dependency that project requires along with the package we install and the nested dependencies.

3) what is package.json & package-lock.json?

→ Package.json:

→ this file lists the direct dependencies (packages) that our project need.

→ It includes package names & version ranges & it automatically updates its version based on '`>`'(carat) or '`~`'(tilde) symbol before it.

* `Carat` - updates minor & patch version

* `Tilde` - updates only patch version.

* if no '`>`' or '`~`' - exact installed version no update happens.

→ Package-lock.json:

→ this file locks the exact versions of the dependencies and their dependencies, ensuring consistent installation across different environments.

eg: "express": "`4.12.3`"
 ↓ ↓
 Major Minor
 Patch

→ the version usually starts at version 1.0.0.

4) what is Nodemon?

→ Nodemon is the utility tool for node.js it automatically restarts the server whenever there is a change.

`npm i -g nodemon` → global installation

→ without Nodemon, whenever we make change we need to stop the server by $\text{ctrl}+\text{c}$ & restart it with node app.js → entry point file.

→ with Nodemon, we can add this to package.json scripts for quick access.

{ Scripts : {

 "start" : "node app.js",

 "dev" : "nodemon app.js"

} }

→ Start the app with npm run dev & it will restart the server every time whenever the file is saved.