1. Introduction to Node.js:

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine.

It allows you to run JavaScript on the server side.

2. Installation:

Download and install Node.js from official website.

3. Create Your First Node.js File:

Create a file named app.js with the following content:

javascript

Copy code

console.log("Hello, Node.js!");

4. Run Your Node.js File:

Open a terminal and navigate to the directory containing app.js.

Run the following command:

bash

Copy code

node app.js

5. Node.js Modules:

Node.js uses a module system. Create a module in a separate file, e.g., myModule.js:

javascript

Copy code

// myModule.js

exports.myFunction = function() {

return "Hello from my module!";

};

Then, in app.js:

javascript

Copy code

const myModule = require('./myModule');

console.log(myModule.myFunction());

6. npm (Node Package Manager):

npm is the package manager for Node.js. Initialize a new project:

bash

Copy code

npm init

Install packages, e.g. Express:

bash

Copy code

npm install express

7. Creating a Simple Web Server with Express:

Install Express:

bash

Copy code

npm install express

Create a file server.js:

javascript

Copy code

const express = require('express');

const app = express();

const PORT = 3000;

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

res.send('Hello, Express!');

});

app.listen(PORT, () => {

console.log(`Server is running on http://localhost:${PORT}`);

});

8. Middleware in Express:

Use middleware functions with Express.

javascript

Copy code

// Example middleware

const logMiddleware = (req, res, next) => {

console.log('Middleware executed!');

next();

};

app.use(logMiddleware);

9. Handling POST Requests:

Use the body-parser middleware to handle POST requests.

bash

Copy code

npm install body-parser

10. Connecting to a Database (e.g., MongoDB):

Use the mongoose library for MongoDB.

bash

Copy code

npm install mongoose

11. Asynchronous JavaScript