



Integration of Backend with Frontend

Course: Server-side Development Using NodeJS, Express and MongoDB

Lecture On: Integration of Backend and Frontend - I

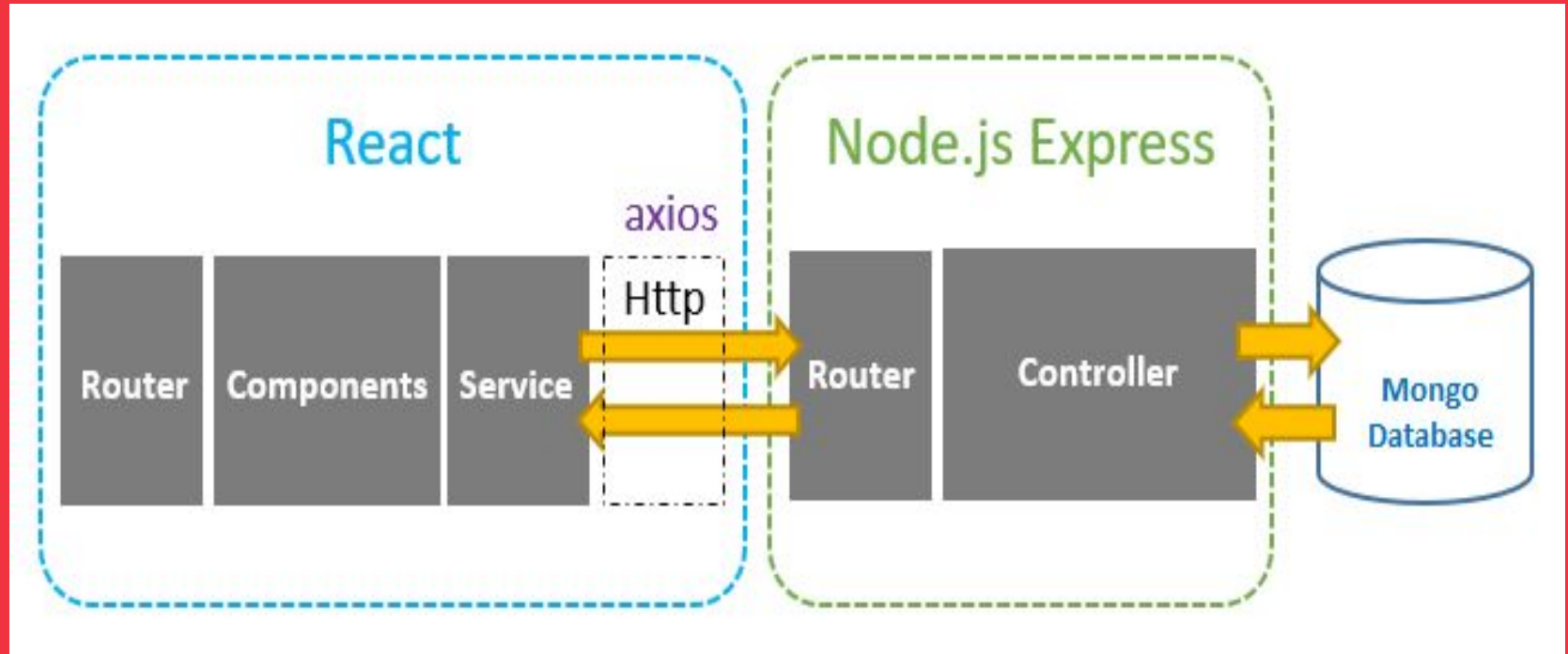
Instructor: Darsh Sanghavi



In the previous class, we covered...

- We developed small application using Node.js, Express.js and MongoDB.
- We also learnt the concept of Multer and understood its applications.

Node-React Framework



Poll 1 (15 sec)

What is the default port for web servers?

1. 1024
2. 80
3. 8080

Poll 1 (Answer)

What is the default port for web servers?

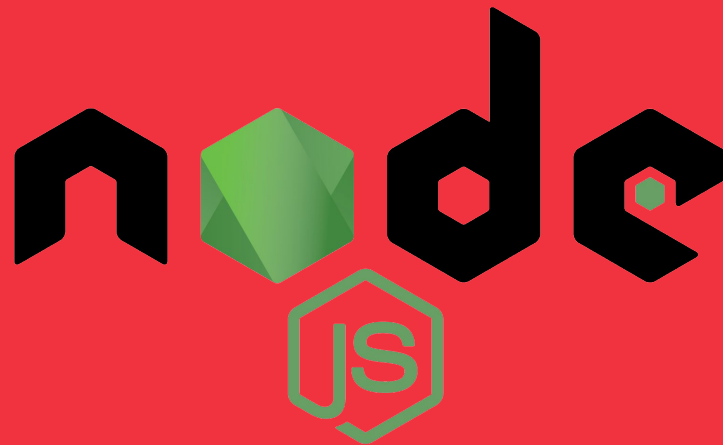
1. 1024
2. 80
3. **8080**

Today's Agenda

- Setting up Node.js project
- Setting up React.js project
- Setting up Environments for Node.Js
- Importance of CORS Middleware
- Introduction to Nodemon
- Connect a basic web-page with a backend API



Setting up Node.js



❏ Step 1:

Go to any director and open the terminal. Then, run the following commands.

```
mkdir node-backend  
cd node-backend  
  
code .
```

❏ Step 2:

Create a file 'index.js' and setup express.

```
const express = require("express");  
const app = express();  
app.get("/", (req, res) => {  
  res.json({ message: "Integration of Node-React." });  
});
```

'**code .**' command is used to open the visual studio code for the newly created folder; however, it is optional.

- ❏ **Step 3:**
Start listening for the requests.

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

Poll 2 (30 sec)

What is 'Express' in Node.js?

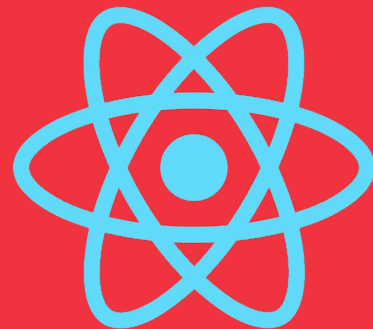
1. NPM Package
2. Web Framework for Node.js
3. Helps is listening to HTTP calls (GET, POST, PUT, etc.)
4. All of the above

Poll 2 (Answer)

What is 'Express' in Node.js?

1. NPM Package
2. Web Framework for Node.js
3. Helps is listening to HTTP calls (GET, POST, PUT, etc.)
4. **All of the above**

Setting up React.js



❏ Step 1:

Go to any director and open the terminal. Then, run the following commands.

```
npm create-react-app react-frontend  
cd react-frontend  
  
npm start  
  
code .
```

'**code .**' command is used to open visual studio code for the newly created folder; however, it is optional.

❏ **Step 2:**

Remove all the boilerplate code that is given by default through the create-react-app command and instead just have a `<div />` with welcome text inside app.js component's return method.

```
return (  
  <div>  
    Welcome to the Frontend with ReactJs.  
  </div>  
);
```

Node.js Environment Variables



- What are Environment Variables in Node.js?
 - Consider environment variables as 'Global' or 'Constant' variables whose values do not change throughout the application's lifecycle.
 - Whenever you need to use certain values that will be common to the entire application, you can declare such values as environment variables.
 - These variables can be used to define values for different environments, such as production environment, beta (development) environment, testing environment and so on.
 - Most commonly used environment variables are PORT no., application's base URL, certain configuration values for DB connectivity, etc.

- ❑ **Step 1:**
Create a .env file. This file should be placed in the root of your project.
- ❑ **Step 2:**
Add the following code inside the .env file that you just created.

```
NODE_ENV = development
PORT = 3000
# Set your database / API connection information here
API_URL=*****
DB_URL="mongodb://localhost:27017/onlineLearning_db"
```

❏ **Step 1:**

Create package.json file using the following command.

```
npm init - y
```

❏ **Step 2:**

Add the following code inside the .env file that you just created.

```
NODE_ENV = development
PORT = 3000
# Set your database / API connection information here
API_URL=*****
DB_URL="mongodb://localhost:27017/onlineLearning_db"
```

❏ Step 3:

Install dotenv from npm.

```
npm install dotenv
```

❏ Step 4:

Finally, now you can read the variable defined in the .env file anywhere in your application.

```
// server.js  
console.log(`Your port is ${process.env.PORT}`); // undefined
```



```
const dotenv = require('dotenv');  
dotenv.config();  
console.log(`Your port is ${process.env.PORT}`); // 3000
```



Poll 3 (15 sec)

What types of variables can be defined in the **.env** file?

1. Local Variables (Example: `var age = 20, userName = "Darsh"`)
2. Global Variables and Constants
3. Only Global Variables

Poll 3 (Answer)

What types of variables can be defined in the `.env` file?

1. Local Variables (Example: `var age = 20, userName = "Darsh"`)
2. **Global Variables and Constants**
3. Only Global Variables

Nodemon



- What is Nodemon? Why should you use it?
 - Nodemon is an NPM package that helps developer speed up the development process.
 - Nodemon is a utility that monitors for any changes in your source and automatically restarts your server.
 - Thus, you do not need to run the 'node index.js' command whenever you make changes in your source code.

❑ **Step 1:**

Install nodemon like any other NPM package.

```
npm install nodemon--save - dev
```

❑ **Step 2:**

Instead of using the 'node server.js' command to run your application, replace it with the following command that will automatically start nodemon service, which will then continuously listen to the changes in the source code.

```
nodemon ./server.js
```

Poll 4 (15 sec)

After installing nodemon, we do not need to run the 'node index.js' command whenever changes are made in the source code.

1. True
2. False

Poll 4 (Answer)

After installing nodemon, we do not need to run the 'node index.js' command whenever changes are made in the source code.

1. **True**

2. False

Homework

- ★ Create one API in NodeJs that takes input of two numbers (num1 and num2)
 - API will be in POST
 - Validate req parameters (num1 and num2 should not be empty)
 - Perform any operation (add, sub, divide) on num1 and num2 and send the answer as response

- ★ Create a webpage with two input fields where user can enter two numbers and one button.
 - A basic `<form />` component that has two `<input />` components and one submit button
 - On pressing the submit button, first validate the form (fields should not be empty)
 - After validation, call the API created in step 1

Doubt Clearance (5 mins)

Key Takeaways

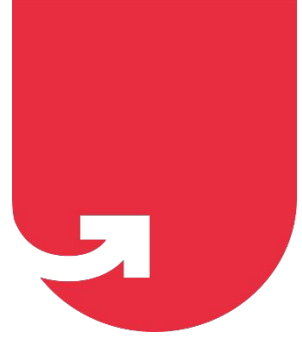
1. We learned how to setup different environments for Node.js (Production/Beta)
2. Importance of CORS package
3. Importance of Nodemon
4. Calling backend APIs from frontend webpage.

The following tasks are to be completed after today's session:

Homework
MCQs
Coding Questions
Course Project - Checkpoint 9

In the next class, we will discuss

- Integrate EdTech project's backend with its frontend



Thank You!