Express Server

Create a simple express server

1. **Express Framework:**

   - **Purpose:** Express is a web application framework for Node.js, designed to simplify the process of building web applications and APIs.
   - **Routing:** Express provides a powerful routing mechanism that allows you to define how your application responds to different HTTP requests (e.g., GET, POST).

2. **HTTP Methods:**

   - **GET:** Used to retrieve data from the server. Typically used for reading information.
   - **POST:** Used to submit data to the server. Often used for creating or updating resources.
   - **Other Methods (PUT, DELETE, etc.):** Used for various purposes, such as updating or deleting resources.

3. **Routes:**

   - **Definition:** Routes define the paths in your application and the HTTP methods they respond to.
   - **Parameters:** Routes can have parameters that allow dynamic handling of different values.

4. **Request and Response Objects:**

   - **Request (`req`):** Represents the incoming HTTP request from the client. Contains information about the request, such as parameters, headers, and body.

- **Response (`res`):** Represents the outgoing HTTP response to be sent back to the client. Allows you to send data, set headers, and more.

5. **Listening and Ports:**

- **Server Creation:** After defining routes and middleware, the Express application needs to be "listened" to for incoming requests.
- **Port:** The server listens on a specific port (e.g., 3000) for incoming HTTP requests.

```
//server.js
// Import the express module
const express = require('express');

// Create an instance of the express application
const app = express();

// Define a route for the root URL ("/")
app.get('/', (req, res) => {
  res.send('Hello, this is the root/main route!');
});

// Define another route for "/api" with JSON response
app.get('/api', (req, res) => {
  res.json({ message: 'This is the API route.' });
});

// Define a route with URL parameters
app.get('/greet/:name', (req, res) => {
  const { name } = req.params;
  res.send(`Hello, ${name}!`);
});

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

Run the server

```
node server.js
```

Visit `http://localhost:3000` in your browser, and you should see the response from the root route. You can also try accessing other defined routes (`/api`, `/greet/:name`).