Node.js Web Server using Express - Beginner's Guide

This document explains how to create a basic web server using Node.js and the Express framework. It covers each part of the code with beginner-friendly explanations and includes clarification for a common doubt about how requests are handled.

Code Explanation

Code:

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

app.get('/', (req, res) => {
  res.send('Hello This is the response i am sending for ur request');
});

const port = 3000;

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

Explanation:

const express = require('express'); – Loads the Express module and assigns it to a variable.

const app = express(); – Creates an instance of the Express application, giving access to methods like .get() and .listen().

app.get('/', (req, res) => {...}) – Defines a route for the root URL '/' that listens only for GET requests. Sends a text response when accessed.

const port = 3000; – Specifies the port number the server will listen on.

app.listen(port, () => {...}) – Starts the server and logs a message to the console indicating it's running.

Understanding HTTP Methods

Different HTTP methods are used for different purposes. Here's a summary:

Method	Purpose

GET	Retrieve data from the server
POST	Send data to the server
PUT	Update existing data
DELETE	Remove data from the server

Clarification to Your Doubt

Your question:

If someone goes to '/' in the website and performs any request (like GET, POST, etc.), will app.get() catch it and send the string response?

Answer:

No, app.get('/') only handles GET requests to the root URL ('/'). It does not handle POST, PUT, or DELETE requests. If someone sends any other type of request to '/', and no other route is defined for it, Express will return a 404 Not Found error.

To handle other types of requests, you can use app.post(), app.put(), app.delete(), etc., or use app.all() to handle all request types.

Examples for Handling Other Methods

```
app.post('/', (req, res) => {
  res.send('This is a POST request to /');
});

app.all('/', (req, res) => {
  res.send(`Received a ${req.method} request at /`);
});
```

This concludes the beginner-friendly explanation for setting up an Express server and understanding request handling.