Date : 25th - Nov- 2020
Morning Session : 9am – 11.00 PM
By ~ Akash

Topics: Node With Express

Detailed Routing :

What is Routing?

Routing defines the way in which the client requests are handled by the application endpoints.

Implementation of routing in Node.js: There are two ways to implement routing in node.js which are listed below:

- By Using Framework
- Without using Framework

Using Framework: Node has many frameworks to help you to get your server up and running. The most popular is Express.js.

Routing with Express in Node: Express.js has an “app” object corresponding to HTTP. We define the routes by using the methods of this “app” object. This app object specifies a callback function, which is called when a request is received. We have different methods in app objects for a different type of request.

For GET request use app.get() method:

```
var express = require('express')
var app = express()
app.get('/', function(req, res) {
   res.send('Hello Sir')
})
```

For POST request use app.post() method:

```
var express = require('express')
var app = express()
app.post('/', function(req, res) {
   res.send('Hello Sir')
})
```

For handling all HTTP methods (i.e. GET, POST, PUT, DELETE etc.) use app.all() method:

```
var express = require('express')
var app = express()
app.all('/', function(req, res) {
   console.log('Hello Sir')
   next()   // Pass the control to the next handler
})
```

The **next()** is used to hand off the control to the next callback. Sometimes we use **app.use()** to specify the middleware function as the callback.

So, to perform routing with the Express.js you have only to load the express and then use the app object to handle the callbacks according to the requirement.

Routing without Framework: Using the frameworks is good to save time, but sometimes this may not suit the situation. So, a developer may need to build up their own server without other dependencies.

Now create a file with any name using .js extension and follow the steps to perform routing

Here we will use the built in module of node.js i.e. http. So, First load http:

```
var http = require('http');
```

Now create server by adding the following lines of code:

```
http.createServer(function (req, res) {

   res.write('Hello World!'); // Write a response

   res.end(); // End the response

}).listen(3000, function() {
   console.log("server start at port 3000"); // The server object listens on port 3000
});
```

Now add the following lines of code in above function to perform routing:

```
var url = req.url;

if(url ==='/about') {
   res.write(' Welcome to about us page');
   res.end();
 } else if(url ==='/contact') {
   res.write(' Welcome to contact us page');
   res.end();
 } else {
   res.write('Hello World!');
   res.end();
}
```

Express Router:

The **express.Router()** function is used to create a new router object. This function is used when you want to create a new router object in your program to handle requests.

Syntax:

express.Router([options])

Optional Parameters:

- case-sensitive: This enables case sensitivity.
- mergeParams: It preserves the req.params values from the parent router.
- strict: This enables strict routing.

Return Value: This function returns the New Router Object.

Express.js express.Router() Function - GeeksforGeeks

Render:

The res.render() function is used to render a view and sends the rendered HTML string to the client.

Syntax: res.render(view [, locals] [, callback])

Parameters: This function accepts two parameters as mentioned above and describes

Locals: It is basically an object whose properties define local variables for the view.

Callback It is a callback function. **Returns:** It returns an Object.****

Express.js res.render() Function - GeeksforGeeks

Express Static :

The express.static() function is a built-in middleware function in Express. It serves static files and is based on serve-static.

Syntax:

express.static(root, [options])

Parameters: The root parameter describes the root directory from which to serve static assets.

Return Value: It returns an Object.

Express.js express.static() Function - GeeksforGeeks

EJS:

EJS or Embedded Javascript Templating is a templating engine used by Node.js. Template engine helps to create an HTML template with minimal code. Also, it can inject data into HTML template at the client side and produce the final HTML. EJS is a simple templating language which is used to generate HTML markup with plain JavaScript. It also helps to embed JavaScript to HTML pages.

Use EJS as Template Engine in Node.js - GeeksforGeeks

For Code explanation please go through the Recorded Lecture