

NODE.JS

Express Router

Sreekanth M. E.

Freelance Trainer & Consultant

<http://www.SreekanthME.com>

Routing refers to the definition of application end points (URIs) and how they respond to client requests.

Route paths, in combination with a request method, define the endpoints at which requests can be made.

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

Route methods are derived from HTTP verbs.

```
app.method (path, handler);
```

app → Instance of express.

method → HTTP request method (like GET, POST, DELETE, PUT, PATCH etc.), in lowercase.

path → Path on the server.

handler → Callback function executed when the route is matched.

Some route methods are:

`app.get()` → Corresponds to HTTP GET

`app.post()` → Corresponds to HTTP POST

`app.delete()` → Corresponds to HTTP DELETE

`app.all()` → Special method that catches all HTTP verbs

ROUTE PATHS

Route paths can be strings, string patterns, or regular expressions.

Query strings are not part of the route path.

Examples of route paths based on string patterns:

`/ab?cd` → Will match `acd` and `abcd`.

`/ab+cd` → Will match `abcd`, `abbcd`, `abbbcd`, and so on.

`/ab*cd` → Will match `abcd`, `abxcd`, `ab123cd`, and so on.

`/ab(cd)?e` → Will match `/abe` and `/abcde`.

`/admin/*` → Will match all paths under "admin".

ROUTE PARAMETERS

Route parameters are named URL segments that are used to capture the values specified at their position in the URL.

The captured values are populated in the **request.params** object, with the name of the route parameter specified in the path as their respective keys.

Route Path: /users/:userId/books/:bookId

Request URL:

<http://localhost:3000/users/34/books/8989>

req.params: { "userId": "34", "bookId": "8989" }

Hyphen (-) and the dot (.) are interpreted literally, they can be used along with route parameters for useful purposes.

Route Path: /flights/:from-to

Request URL: <http://localhost:3000/flights/LAX-SFO>

req.params: { "from": "LAX", "to": "SFO" }

Route Path: /plantae/:genus.:species

Request URL:

<http://localhost:3000/plantae/Prunus.persica>

req.params: { "genus": "Prunus", "species": "persica" }

Route handlers (callbacks) can be in the form of:

- A Single function,
- A comma separated list of functions
- An array of functions
- Various combinations of all the above.

APP.ROUTE()

Chainable route handlers for a route path can be created by using `app.route()`.

Because the path is specified at a single location, chainable routers reduce redundancy and typos as well as make code more readable.

EXPRESS.ROUTER()

`express.Router` class can be used to create modular, route handlers.

A Router instance is a complete middleware and routing system; for this reason, it is often referred to as a “mini-app”.

Middlewares can be mounted directly on the router.

Note: We will learn more about “middlewares” in the next chapter.

END OF CHAPTER

APPENDIX