

# Tutorial @4

Express routing options (easy)

# What is routing?

Routing is a method that refers to determining how an application responds to a client request to a particular path and a specific HTTP request method (GET, POST, and so on).

In simple terms, routing is controlling which function gets invoked whenever the user navigates to a particular URL. In this context, URL refers to any path or route.

# Defining routing methods

We can define routing methods like so:

`app.METHOD(PATH, CALLBACK)`



`'/', '/about', '/posts/:slug'`

`function (req, res) { ... }`

Most commonly used verbs in place of **METHOD** are:

<b>get</b>	To handle GET requests (i.e. to request/GET data from a specified resource).
<b>post</b>	To send data to a server to create/update a resource.
<b>put</b>	To send data to a server to create/update a resource. The difference between POST and PUT requests is that the latter are idempotent. This means that PUT requests have no additional effect if they are called multiple times. In contrast, if you call a POST method more than once, your program will have side effects. Therefore, bear in mind that POST requests are not to be called more than once.
<b>delete</b>	For removal of a specified resource.

*Source*

# app.all

There is a special routing method, `app.all()`, used to load middleware functions at a path for all HTTP request methods.

```
14 // app.all example
15 app.all('/students', (req, res, next) => {
16     console.log(`${ req.method } : ${ req.url }`)
17     next()
18 })
19
```

# app.route

You can create chainable route handlers for a route path by using `app.route()`. Because the path is specified at a single location, creating modular routes is helpful, as is reducing redundancy and typos.

```
20 // app.route example
21 app.route('/students')
22   .get((req, res) => {
23     res.sendFile(getView('students'))
24   })
25   .post((req, res) => {
26     res.send('Post request...')
27   })
28
```

# express.Router

Use the `express.Router` class to create modular, mountable route handlers. A Router instance is a complete middleware and routing system; for this reason, it is often referred to as a “mini-app”.

```
1  const express = require('express')
2  const router = express.Router()
3
4  router.get('/', (req, res) => {
5    res.send('Tutors page...')
6  })
7
8  router.get('/:name', (req, res) => {
9    res.send(`You are viewing ${ req.params.name }'s page...`)
10 })
11
12 module.exports = router
13
```

root/routes/tutors.js

```
6  const tutors = require('./routes/tutors.js')
7  app.use('/tutors', tutors)
```

root/app.js

>>See you