

# EXPRESS

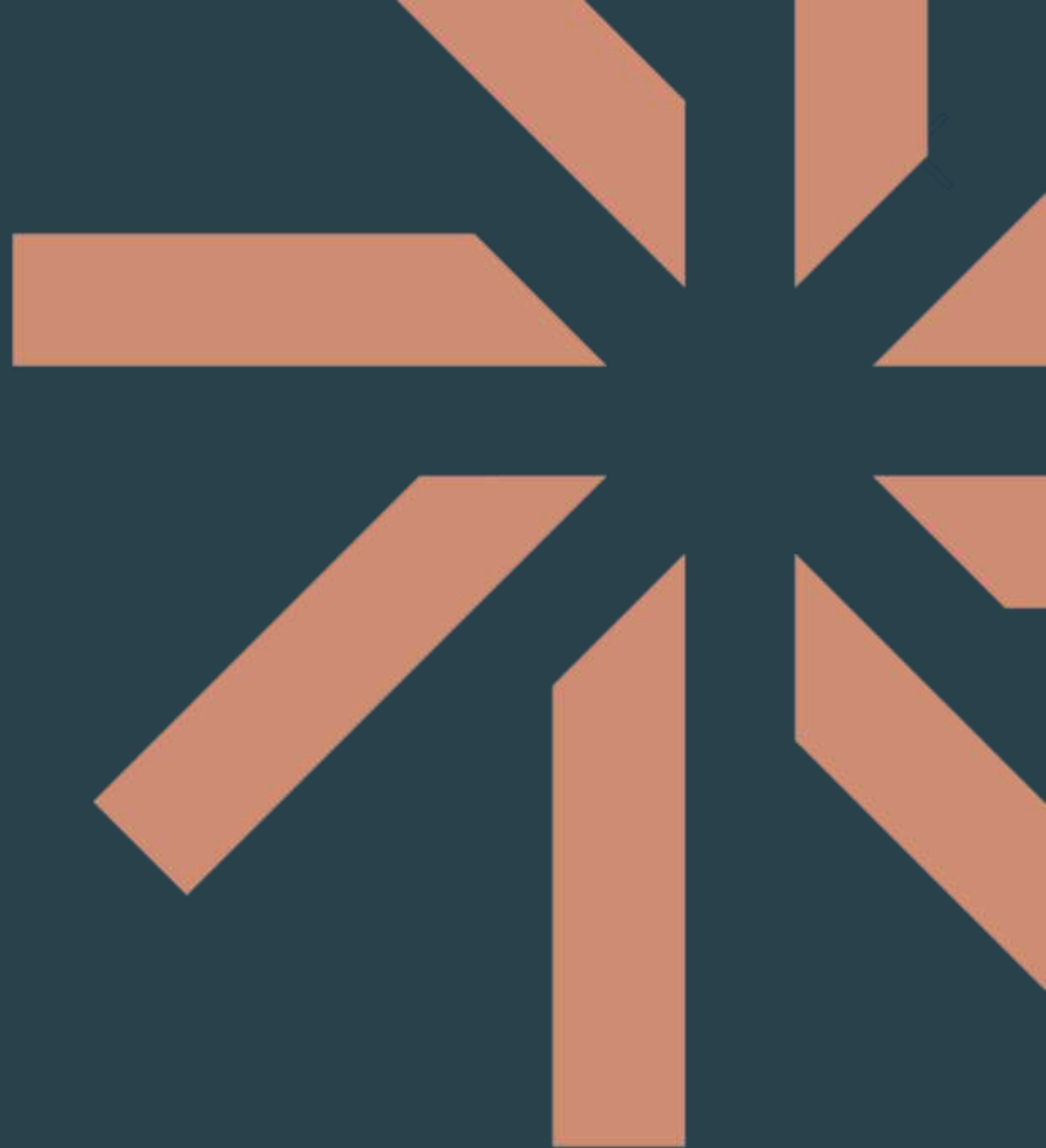


Qinshift   
Academy

# WHAT IS EXPRESS.JS?

Express.js, or simply Express, is a back end web application framework for building RESTful APIs with Node.js, released as free and open-source software under the MIT License.

Express.js is one of the best backend development JavaScript Frameworks. The primary usage of it is creating Restful API's what accept request from frontend and send the appropriate response.



# Why Express.js?



- You only need to learn one language for both frontend and backend development, Which is JavaScript. To develop a web frontend, You need HTML, CSS, and JS. So, JavaScript is necessary and the same language is used to code backend if it's Express.js.
- The support of Node.js is one of the best reasons because of which you should choose Express.js. So that you can develop an Express.js backend with the support of all the NPM packages that make the process easier.
- You would be able to debug your application faster than ever with the help of the debugging mechanism it offers to its users. This helps to find error points easily without taking much time.
- While Express itself is fairly minimalist, developers have created compatible middleware packages to address almost any web development problem. There are libraries to work with cookies, sessions, user logins, URL parameters, POST data, security headers, and many more.

# Pros & Cons



## Create a simple server



```
const express = require("express");  
const app = express();  
const port = 3000;
```

The first two lines `require()` (import) the `express` module and create an Express application. This object, which is traditionally named `app`, has methods for routing HTTP requests, configuring middleware, rendering HTML views, registering a template engine, and modifying application settings that control how the application behaves.

## Create a simple server



```
app.get("/", function (req, res) {  
  res.send("Hello World!");  
});
```

The middle part of the code shows a route definition. The `app.get()` method specifies a callback function that will be invoked whenever there is an HTTP GET request with a path (`/`) relative to the site root. The callback function takes a request and a response object as arguments, and calls `send()` on the response to return the string "Hello World!".

## Create a simple server



```
app.listen(port, function () {  
  console.log(`Example app listening on port ${port}!`);  
});
```

The final block starts up the server on a specified port ('3000') and prints a log comment to the console. With the server running, you could go to localhost:3000 in your browser to see the example response returned.



## Create route handlers



Often it is useful to group route handlers for a particular part of a site together and access them using a common route-prefix (e.g. a site with a Wiki might have all wiki-related routes in one file and have them accessed with a route prefix of `/wiki/`). In Express this is achieved by using the `express.Router` object. For example, we can create our wiki route in a module named `wiki.js`, and then export the Router object, as shown in the picture (in the following slide)



# Create route handlers



```
// wiki.js - Wiki route module

const express = require("express");
const router = express.Router();

// Home page route
router.get("/", function (req, res) {
  res.send("Wiki home page");
});

// About page route
router.get("/about", function (req, res) {
  res.send("About this wiki");
});

module.exports = router;
```

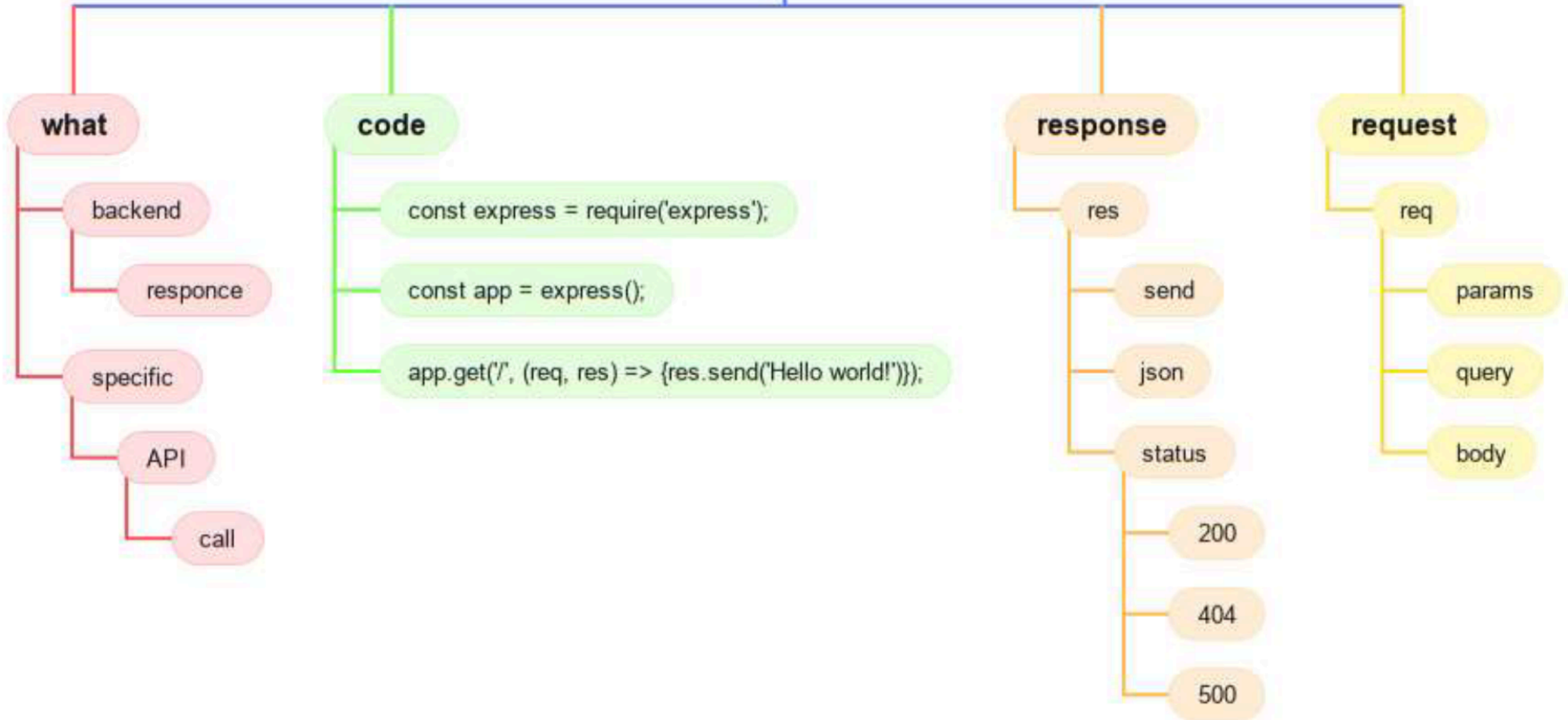
## Create route handlers



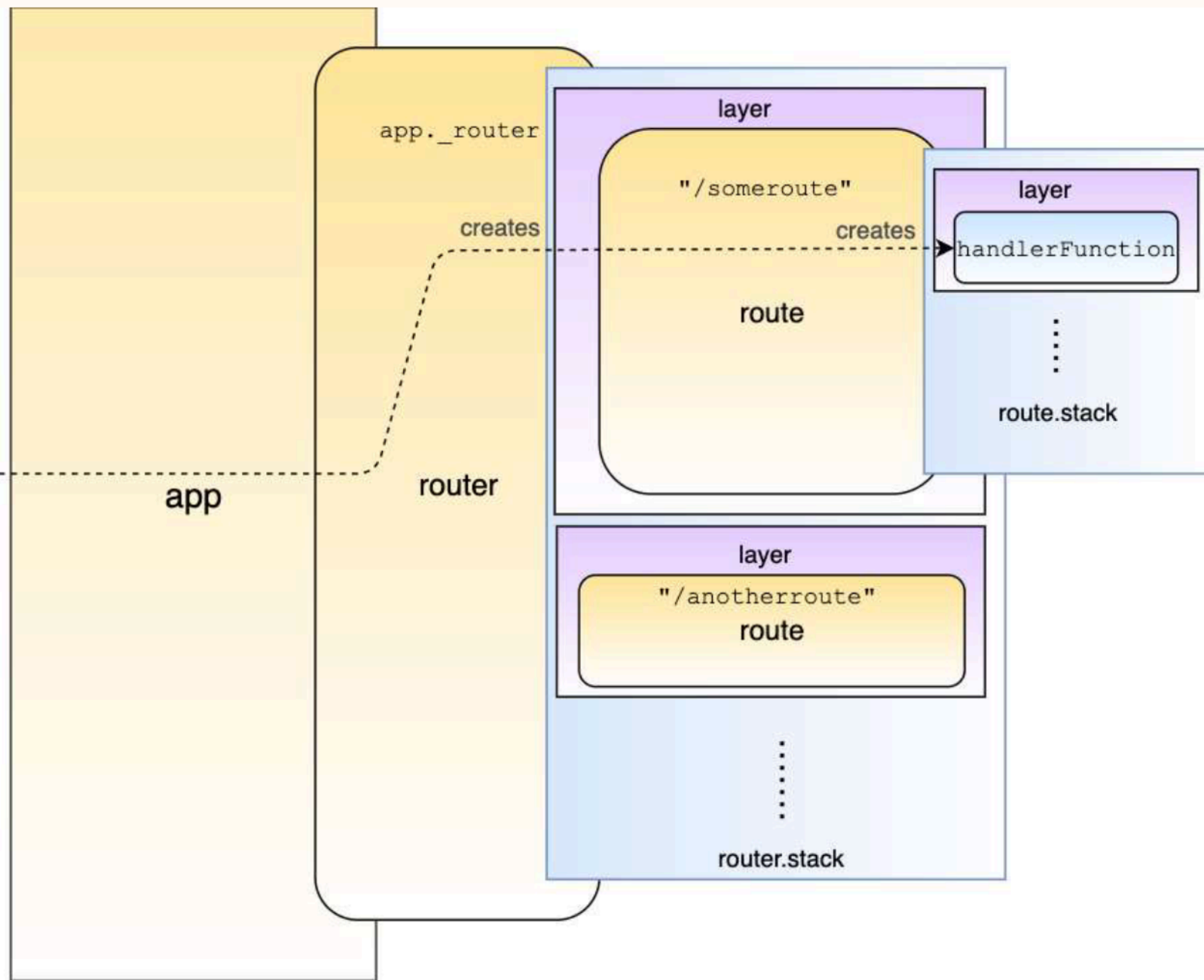
To use the router in our main app file we would then `require()` the route module (`wiki.js`), then call `use()` on the Express application to add the Router to the middleware handling path. The two routes will then be accessible from `/wiki/` and `/wiki/about/`.

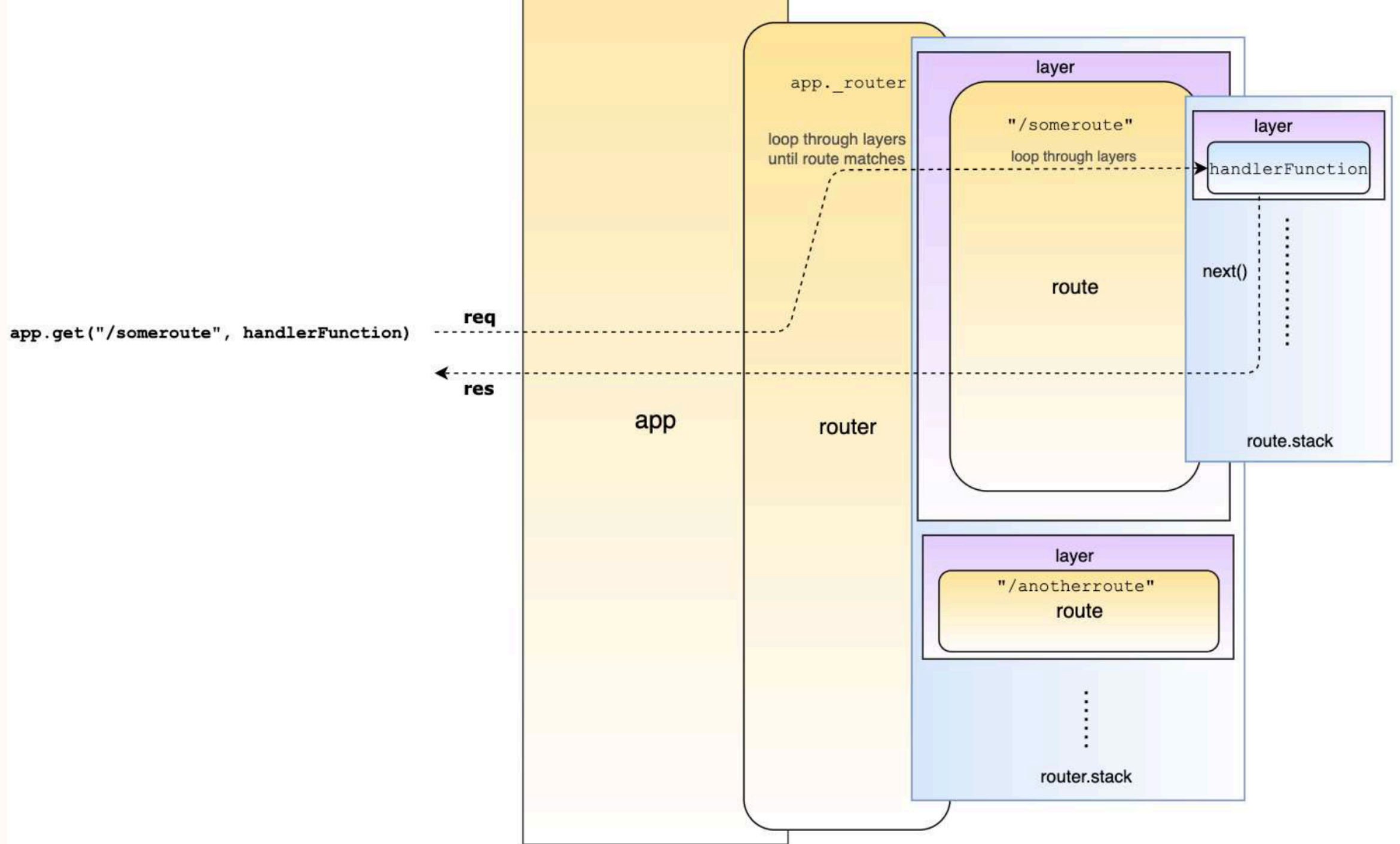
```
const wiki = require("./wiki.js");  
  
// ...  
  
app.use("/wiki", wiki);
```

# express\_js\_routing



`app.get("/someroute", handlerFunction)`





# QUESTIONS?

You can find us at:

[anetastankovskaane@gmail.com](mailto:anetastankovskaane@gmail.com)

[igorveic7@gmail.com](mailto:igorveic7@gmail.com)

