**Defining Routes with Express JS for your App -** [sadafzahra](https://sadafzahra1223.medium.com/?source=post_page-----f1fc2a1b1d41--------------------------------) Jun 2, 2022

Before starting to write code for your application, taking some time to think about its basic architecture helps to give it a good structure to work with.

Most Front-end developers will start to write code for their application from the browser. But it helps to know how to setup a simple back-end to handle page routing and also allowing the browser to make requests like API calls.

In this article, we explore how to setup simple routes. Each route defined will load a new page when accessed as a URL in the browser.

Before your begin, make sure you have installed Node JS for your computer’s operating system. Use the following prompt to check if it is already installed or not.

$ node -v

First, setup a project directory. I’m using the Linux command line so I can use the following command to do this.

$ mkdir *project-name*

Don’t forget to change directory to your project with $ cd *project-name .*Then initialize your project with npm which is the package manager for node and will install project dependencies like expressjs and setup scripts to test in the browser (this is further described below).

$ cd *project-name*  
$ npm init //I accepted all the default values by pressing enter

After initializing your project with npm, you will see a package.json file appear in your project directory. Now install express with the following command. Once installed, you should see the package appear under dependencies in the package.json file.

$ npm install express --save

Create a file called index.js in your project directory and add the following code.

const express = require('express')  
const app = express()app.get('/', function (req, res) {  
 res.send('Home Page')  
})app.get('/page1', function (req, res) {  
 res.send('page1')  
})app.listen(3000)

Above we created an app object that uses the express js framework. We used the get() method available on the object to **define a route** and send a response to the browser that requested the page. We defined two routes / for the home page and /page1 for any other route that you want to define.

Finally, listen() method on the app will allow you to access the defined routes from the browser’s address bar. But first you have to run the index.js file like this.

$ node index.js

Or in your package.json file your could add the following script and type npm start in your command prompt to run the script.

"scripts": {  
 "test": "echo \"Error: no test specified\" && exit 1",  
 "start": "node index.js"  
 },

Access the page routes you defined from the browser with the following.

[http://localhost:3000](http://localhost:3000/)/[http://localhost:3000](http://localhost:3000/)/page1