<https://alejandronapoles.com/2016/03/12/the-simplest-webpack-and-express-setup/>

<https://scotch.io/tutorials/use-expressjs-to-deliver-html-files>

<https://scotch.io/tutorials/getting-started-with-webpack-module-bundling-magic>

How to use nginx?

SASS - Sass is a [scripting language](https://en.wikipedia.org/wiki/Scripting_language) that is [interpreted](https://en.wikipedia.org/wiki/Interpreted_language) or [compiled](https://en.wikipedia.org/wiki/Compiled_language) into [Cascading Style Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets) (CSS)

What compiles / interprets it?

Sass-loader

npm install node-sass sass-loader --save-dev

if you want your css dynamically injected:

npm install style-loader css-loader --save-dev

The path meanings:

* / is the root of the current drive; this is an absolute path, beginning in the root of the file stystyle and navigating hrough the folders given by name.
* ./ is the current directory; a relative path name. start with current working directory and apply the navigation operations.
* ../ is the parent of the current directory.
* Node.js > \_\_dirname – is a global variable that gives you the path of the current running file.
* The path inside require is always relative to the file containing the call to require.
* ./ gives you directory from which you ran the node command

EXPRESS

Express() – this is a module that facilitates..

Often it is assigned to an ‘app’ variable

Var app = express();

Then, methods are used on app:

App.get(path, callback) – routes http get requests to the specific path

STATIC FILES – scripts, images, CSSfiles, etc. They aren’t server-generated, but are sent to the browswer when requestsed. In express, we must configure it to serve the static content.

Express.static(x) – this takes an argument that specifies the root directory from which to serve your statuic files. This way, we must tell express where to find these files.

Webpack – a bundler

INSTALLATIONS

npm i express **-**S

npm i webpack **-**D

FOLDERS AND FILES

1. dist – client code 🡪
   1. Bundle.js – this is the bundled js files loaded by index.html
2. src – save production / dev files ->
   1. index.html
   2. script.js
3. node\_modules

FILES

1. Webpack.config.js – this is the webpack controller.
   1. It details the entrypoint, the directory,
   2. It creates bundle.js
   3. It also defines loaders, such as babel
      1. It installs presets -
2. Package.json –
   1. Identifies the project
   2. Script – a list of npm commands
      1. Test – npm test
      2. Npm start
   3. this lists dependencies to be installed during npm install.
3. Server.js – this serves the

console commands

webpack --- this creates a bundle.js in the place designated by webpack.config.js

npm run dev –

this runs the the commands defined in scrips.dev of package json.

Npm scripts start servers and run tests

devDependencies – list your tests.

Custom scripts:

"scripts": {

"script1": "node script1.js"

}

to my package.json file but when I try running script1 I get the following output:

npm run script1

SRC – files that need to be manipulated. JS files that will be transformed by webpack.

DIST – files that are ready to go – index.html, which serves up bundle.js