8. Web Servers

Github Repo

Today's menu

- HTTP Review
- Basic web server review
- Simple routing with HTTP
- Better routing with Express
- What are middlewares?
- Using a rendering engine

JSON VIEWER :D

https://chrome.google.com/webstore/detail/json-viewer/gbmdgpbipfallnflgajpaliibnhdgobh

HTTP Review

- HTTP runs on top of TCP
- HTTPS is conceptually the same, except that the content is encrypted by TLS
  - It's transparent at the HTTP level
- Anatomy of an HTTP Request:

```
GET /maps HTTP/2.0
User-Agent: Chrome 80
Cookie: chocolate chips
```

- Anatomy of an HTTP Response:

```
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: text/html; charset=utf-8
Content-Length: 14
Date: Mon, 20 Jan 2020 17:07:16 GMT
Connection: keep-alive

<html><body><!-- Some HTML content... --></body></html>
```

200 -> All good 300 -> Redirect 400 -> Asking for something that you don't have access to 500 -> Server is sad

Basic Web Servers

You can build a basic web server with the http package included in Node.js, however it is unpractical for most uses. Compare the routing mechanisms of the simple_routing.js file with express_server.js file to appreciate the difference.

- One callback function handles everything
- Rarely used in practice
- View simple_routing.js to see how it's built

Intro to Express

Express is a framework built for Node.js, and it's main purpose is to manage routes easily and add quick and easy support for middleware.

Example route:

```
app.get("/hello", (req, res) => {
  res.send("Hello world!");
});
```

What is "middleware?"

Middleware functions are the perfect place to modify the req and res objects with relevant information. Its functions that helps us manage, or modify req and/or res objects.

Example:

```
app.use((req, res, next) => {
  if(req.header('user-agent') === 'Me') {
    res.send('Please use a browser');
  } else {
    next();
  }
});
```

Server side rendering with EJS templates

EJS stands for Embedded JavaScript Templates are used to avoid stringing a bunch of HTML strings together when sending a response. To solve that problem, rendering the HTML of a page from the server side using a language made for templating create an HTML version of your dynamic content before sending it in the response.
Calling `res.render(template, variables)` will use a template. Populating `variables` in this case would be an object containing all the variables you want to access in your template code.
<% %>: for logic, non-outputing <%= %>: will output some javascript
Nodemon - Restarts the node application when file changes in the directory are detected. You must specify this in the package.json
EJS Express

View on Compass | Leave Feedback