CS 4220

- Current Trends in Web Design & Development -

Prof. Cydney Auman

AGENDA

01 Review from Tuesday

02 Using Express Middleware

03 Using Express Router

04 Code Demo

Flashback to Tuesday 4/13/21

- HTTP Verbs - (GET, POST, PUT, DELETE)
- HTTP Path and Query Parameters
- HTTP Status Codes (200, 201, 404, 500)
- CORS - Cross-Origin Resource Sharing
  - Mechanism that uses additional HTTP headers to to access resources from a server located at a different origin.
- REST architecture
  - Client-Server Separation
  - Stateless Constraint
  - Cache Constraint
  - Uniform Interface

Express Middleware

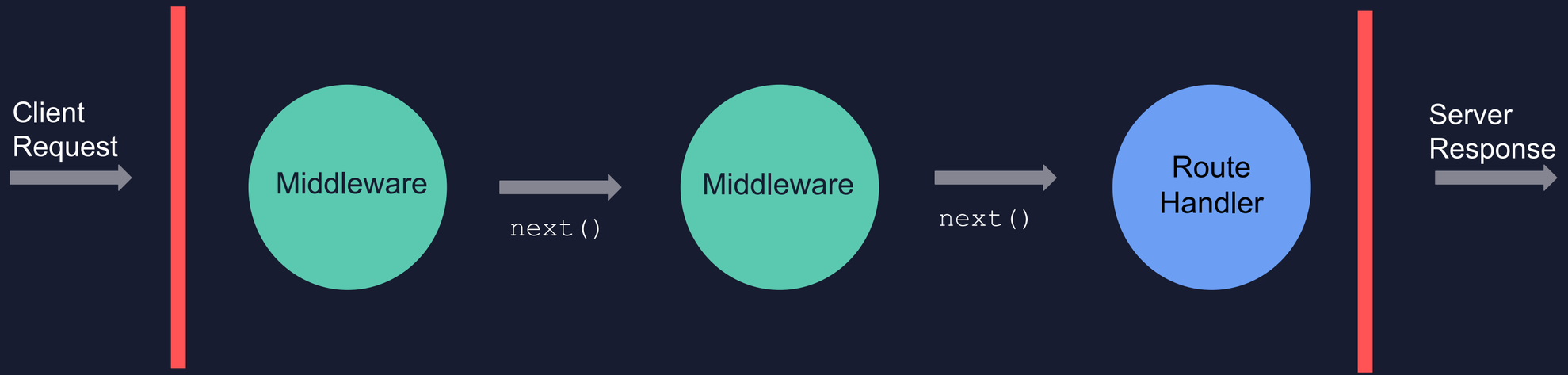
Middleware are functions that have direct access to both the request object and the response object. Middleware also has a special function typically referred to as next. This next function has access to the next process in request-response life cycle.

Middleware functions can perform a variety of tasks:

- Execute code.
- Perform changes to the request and the response objects.
- Execute a call to the next middleware.
- End the request-response cycle as needed.

How Middleware Works

When a request is made to our Node.js Server - the Middleware function will receive the Request and Response objects *before* the Routes. Inside Middleware functions various tasks can be performed. Authentication, Data Validation, Body Parsing, Logging and much more. Once this code is done executing inside the Middleware, the next function must be called. By calling the next function the Request and Response objects will be sent to the next Middleware function if present and else it is sent to the Route handlers.



Express Middleware Levels

Application Level Middleware

These are Middleware functions that are bound to the instance of the `app` object. Application Level Middleware in Express.js is applied by calling `app.use()` and passing in the Middleware function.

Router Level Middleware

These are Middleware functions that are only bound to the Router level instead of the entire app. Router Level Middleware in Express.js is applied by calling `router.use()` and passing in the Middleware function.

Express Router

In previous lectures we covered Routing, how our Node.js Server endpoints respond to client requests.

An **Express Router** is an object that represents an isolated instance of routes and middleware. It does not have the full functionality and feature of the Express App thus it is only capable only of performing these routing and middleware function.

The Express Router provides a way to create a collection of related routes. This allows for better structure and organization in the Node.js Server code.

Accessing the Express Router is simple and concise:

```
const router = require('express').Router();
```

```
console.log(‘Week 12’);
console.log(‘Code Examples’);
```

Review and Prep

Review

- Review Slides & Run Code Examples
- Read more about Express Middleware
  https://expressjs.com/en/guide/using-middleware.html
- Read more about Express Router
  https://expressjs.com/en/4x/api.html#router

QUIZ

- Opens: 4/15/21 @ 7:30PM
- Closes: 4/15/21 @ 11:00PM
- 10 Questions: Multiple Choice - True/False - Fill in Blank
- Time Limit: 11 minutes (once started)