

# Express

Lesson 4 Review



# Promises

- ◆ Promises wrap code in an API that allows us to trigger resolve or reject events at certain points within our code
- ◆ Promises primarily utilize the 'then' method to fire off an operation once it has completed
- ◆ A sequence of promises can be promised before firing the 'then' method as well



# File System (fs)

- ◆ The File System (fs) module allows us to interact with our apps file system
- ◆ We can both read and write using 'fs'
- ◆ If no encoding arg is passed, the fs will read data as a raw binary stream



# Middleware

- ◆ Glues together complex systems
- ◆ Can act as a interpreter between two incompatible systems



# Express Middleware

- ◆ Interprets user requests and executes an appropriate action
- ◆ Connects our model to our controller
- ◆ Connects our controller to our view



# Express

- ◆ A Node.js web server project
- ◆ Just some helpful sugar on top of Node.js  
`createServer`
- ◆ Provides declarative routing



# Scaffolding

- ◆ The act of stubbing out predefined template blocks
- ◆ Scaffolding can create application folders, commonly used files, configurations, presentation files, as well as predefined package files for libraries
- ◆ Very much opinion based



# Express Generator

- ◆ Scaffolds an Express Application with common folders and configurations
- ◆ Allows us to define a template engine and a CSS preprocessor engine, as well as many other details during our bash call



# SCSS/SASS

- ◆ CSS Preprocessor language
- ◆ Requires transpiling
- ◆ Can have extra build steps including auto-prefixing and gzipping
- ◆ Allows for nested selectors
- ◆ Allows for mixins (functions) and variables



# EJS

- ◆ Simple JavaScript template language
- ◆ Inline with standard HTML tags
- ◆ Uses a delimiter set to isolate JavaScript operations (`<%= %>`, `<% %>`)
- ◆ Parses JavaScript inline with the HTML
- ◆ Requires transpiling