

My Home Slice Service Layers Re-Design

Maritess Manese

SWDV 691: Capstone

Professor Joseph Gradecki

March 28, 2020

My Home Slice Service Layers

Express is a lightweight framework for Node.js. The main goal of Express is to build web applications. Express is an NPM package loaded with methods and optional plugins to build web applications and APIs.

Express listens to incoming requests and figures out what is being asked from the browser. Then Express builds a response and sends that back to the client. One of the most critical things Express does is help start up a server to listen to requests. Express will then parse the incoming requests, turn requests into objects, and craft an HTTP response code and associated content.

Express creates a JavaScript request object (req) automatically by parsing the incoming HTTP request information and then passes it in as the first argument to this callback. Then there is a response object (res) that will send and generate an HTTP response.

The app object conventionally denotes the Express application. Create it by calling the top-level Express () function exported by the Express module:

```
JS index.js
FirstApp > JS index.js > ...
1  const express = require("express");
2  const app = express();
3
4  // '/' is the root path,
5  // req is the request object
6  // res is the response object that will send the http response "Welcome to the home page!"
7
8  app.get('/', (req, res) => {
9    res.send('Welcome to the home page!')
10 })
11
12 app.listen(8080, () => {
13   console.log("LISTENING ON PORT 8080")
14 })
15
```

This basic example demonstrates the `res.send('Welcome to the home page!')` will display on localhost 8080/.

The app object has methods for:

- Routing HTTP requests.
- Configuring middleware
- Rendering HTML views
- Registering a template engine

Express also has properties that affect how the application behaves.

Here are some examples of the request and response messages that My Home Slice users may encounter:

UserAction	Request	Situation	Response
Add recipe	Request Add Recipe Page	Successfully found Add recipe page	HTTP display Add Recipe Page
	Request Add Recipe Page	Could not find Add Recipe Page	Error Add Recipe Page could not be found

UserAction	Request	Situation	Response
Delete recipe	Request to delete recipe	Warning to request to delete recipe	HTTP display Are you sure you want to delete recipe? Yes/no
	Request to delete recipe	Successfully deleted recipe	HTTP display Recipe Successfully Deleted
	Request to delete recipe	Could not Delete recipe	HTTP display Error could not delete recipe at this time

UserAction	Request	Situation	Response
Save Recipe	Request to Save Recipe	Successfully Saved recipe	HTTP display recipe Saved!
	Request to Save Recipe	Could not save recipe	Error Recipe Not Saved!

UserAction	Request	Situation	Response
Edit Recipe	Request to Edit Recipe	Successfully found Edited recipe page	HTTP display Recipe Edit Page!
	Request to Edit Recipe	Could not find Edit Recipe	Error cannot edit recipe at this time!

Here are some examples of the request and response messages for My Home Slice Stretch

Features:

Keyword Search Feature in Nav Bar

UserAction	Request	Situation	Response
Search keywords for recipe	Request list of recipes that contain keyword	Successfully find recipes that have Keywords	HTTP display list of recipes that contain keyword
	Request list of recipes that contain keyword	Could not find recipes that contain keyword	HTTP display There are no recipes that contain keyword

Picture Gallery

UserAction	Request	Situation	Response
Add picture	Request Add picture	Successfully Added picture	HTTP displays the added picture
	Request Add picture	Could not Add picture	Error picture cannot be added at this time

UserAction	Request	Situation	Response
Delete picture	Request to delete picture	Warning to request to delete picture	HTTP display Are you sure you want to delete picture? Yes/no
	Request to delete picture	Successfully deleted picture	HTTP display picture Successfully Deleted
	Request to delete picture	Could not Delete picture	HTTP display Error could not delete picture at this time

Mongoose is an Object Data Mapper that provides the data models of the application data and defines a schema. Mongoose is an easy way to validate data and build complex queries from JavaScript.

Here is an example of a Mongoose products model for a produce stand:

A screenshot of a code editor window titled 'product.js'. The code defines a Mongoose schema for a product model. It starts by requiring the mongoose module, then creates a new mongoose.Schema object with three fields: 'name' (String, required), 'price' (Number, required, with a minimum of 0), and 'category' (String, lowercase, with an enum of 'fruit', 'vegetable', and 'dairy'). The schema is then used to create a mongoose model named 'Product', and the model is exported as the module's exports.

```
1  const mongoose = require('mongoose');
2
3  const productSchema = new mongoose.Schema({
4    name: {
5      type: String,
6      required: true
7    },
8    price: {
9      type: Number,
10     required: true,
11     min: 0
12   },
13   category: {
14     type: String,
15     lowercase: true,
16     enum: ['fruit', 'vegetable', 'dairy']
17   }
18 })
19
20 const Product = mongoose.model('Product', productSchema);
21
22 module.exports = Product;
```

Express, EJS, Mongoose, MongoDB and Nodejs is the stack that is used to build the My Home Slice web application. Express is the Framework used to build the application. EJS works together with Express to create Dynamic HTML with Templating. Mongoose models the data, and MongoDB is the Database.