

Recipe Recommender Source Documentation

Backend Source files

index.js

This component establishes the MongoDB connection.

server.js

This component exposes all the API endpoints on the server at a particular port number.

.env

This component contains the DB uri, port number to connect and Gmail address

recipesDAO.js

This component contains a function to retrieve the recipes based on ingredients.

Functions:

1. injectDB
 - a. Param - conn
 - b. Return - none
 - c. Functionality - connects to the Mongo database
2. getRecipies
 - a. Param - none
 - b. Return – recipesList, totalNumRecipies
 - c. Functionality – Retrieve the recipes based on the ingredients
3. getCuisines
 - a. Param - none
 - b. Return – RecipeList and totalNumRecipes
 - c. Functionality –
 - i. Stores data from the filters variable
 - ii. Creates regex search string for querying the mongo database based on the user's inputs
 - iii. Stores the query results and formats them
 - iv. If user checked the 'send email' checkbox, sends an email to them containing their recommended recipes using nodemailer
4. addRecipe
 - a. Param - recipe
 - b. Return - Status
 - c. Functionality -
 - i. Copies data from recipe to inputRecipe and maps to appropriate keys
 - ii. Creates string of ingredients, restaurants, locations

- iii. Stores the inputRecipe dictionary in the database
 - iv. Returns the Status
- 5. getIngredients
 - a. Param - none
 - b. Return - Ingredients list or error
 - c. Functionality -
 - i. Extracts all the ingredients from database
 - ii. return the extracted list

recipes.controller.js

This component retrieves the result using the functions in recipesDAO

Functions:

1. apiGetRecipes
 - a. Param - req, res, next
 - b. Return - RecipeList
 - c. Functionality - calls recipeDAO and passes it the filters variable, then retrieves the response containing recipes using getRecipes in recipesDAO
2. apiGetRecipeCuisines
 - a. Param - req, res, next
 - b. Return - cuisines
 - c. Functionality - retrieves the cuisines response using getRecipes in recipesDAO.
3. apiPostRecipe
 - a. Param - req, res, next
 - b. Return - Status
 - c. Functionality - call recipesDAO and passes the request body, and returns the response provided by this recipeDAO call
4. apiGetIngredients
 - a. Param - req, res, next
 - b. Return – ingredients
 - c. Functionality - retrieves the ingredients response using getRecipes in recipesDAO.

recipes.route.js

This component creates a new router object to handle the requests

Frontend Source files

App.js - ReactJS Class

App Component is the main component in React which acts as a container for all other components. This is the root component which calls child component - Form.js & RecipeList.js. Each of these child components return data back to App.js after they process information.

Instance variables:

1. Ingredients - is a set data structure. No duplicate ingredients are allowed
2. recipeList - is array that store all recipe details
3. cuisine – stores the cuisine type input from the user
4. email – stores the email id, if input by the user
5. flag – stores the value of 'checked' of the checkbox on the UI form. If checked equals 'true', an email with the list of recipes is sent to the user.
6. isLoading - variable to check whether data is loading or noIngredientsInput

Functions:

1. getRecipeDetails
 - a. Param - {String} ingredient, cuis, mail, flag
 - b. Return - none
 - c. Functionality - This method gets all recipes from given ingredients by querying MongoDB. Sends the data to the '/recipes' api and starts the backend processes.
2. handleSubmit - handles the stream of actions immediately after clicking 'search recipes'. Calls getRecipeDetails
 - a. Param - {array} formDict – stores the input details from the form
 - b. Return - none
 - c. Functionality - This method is a callback function of Form component which sets instance variable - Ingredients in App.js
3. constructor
 - a. Param - none
 - b. Return - none
 - c. Functionality - initialize default values to instance variables

Form.js - ReactJS Class

Form Component takes user inputs and sends to back to App/Root component

Instance variables:

1. Ingredients - is a set data structure. No duplicate ingredients are allowed
2. Dict - Dictionary that stores all of the input details from the form

Functions:

1. handleSubmit
 - a. Param – event
 - b. Return - none
 - c. Functionality - This method is called by App.js and returns all of the user inputs through the variables dict
2. constructor
 - a. Param - none
 - b. Return - none
 - c. Functionality - initialize default values to instance variables
3. printHandler
 - a. Param - event
 - b. Return - Html list of ingredients
 - c. Functionality - This method displays the ingredients and the recipe instructions after clicking search recipes.
4. addHandler
 - a. Param - event
 - b. Return - none
 - c. Functionality - This method updates ingredients set when adding an ingredient
5. removeHandler
 - a. Param - event
 - b. Return - none
 - c. Functionality - This method removes ingredients from the set upon clicking on the ingredient
6. Render
 - a. Param - none
 - b. Return - none
 - c. Functionality - constructs UI of Form
7. componentDidMount
 - a. Param - none
 - b. Return - none
 - c. Functionality - Calls backend to get ingredients list and sets states of ingredients list and cuisine list

Recipe.js - React Functions

Creates a modal view of a recipe

RecipeCard.js - React Functions

Provides a card view form each recipe

RecipeList.js - React Functions

Creates a list of RecipeCard component of all the recipes received as a input parameter and handles modal view for viewing complete recipe details

Instance variable:

1. isOpen - to check whether a modal view is open to view a recipe
2. currentRecipe - store information about which recipe is currently viewed in modal view

Functions:

1. handleViewRecipe
 - a. Param - recipe card clicked
 - b. Return - none
 - c. Functionality - sets isOpen state as true and currentRecipe variable to input param
2. onClose
 - a. Param - None
 - b. Return - None
 - c. Functionality - unsets isOpen state

TypeAheadDropDown - React Functions

Provides regex type search for Ingredients

Instance variable:

1. Suggestions - state to store ingredients suggestions during search
2. Text - input ingredient text

Functions:

1. onChange
 - a. Param - Event
 - b. Return - None
 - c. Functionality - Using input param to the TypeAheadDropDown component and Text, create suggestions.

