

Recipe Recommender Source Documentation

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.

Functions:

1. getRecipeDetails
 - Param - {String} ingredient, cuis, mail, flag
 - Return - none
 - 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
 - Param - {array} formDict – stores the input details from the form
 - Return - none
 - Functionality - This method is a callback function of Form component which sets instance variable - Ingredients in App.js
3. constructor
 - Param - none
 - Return - none
 - Functionality - initialize default values to instance variables

4. postRecipeDetails
 - Param - {String} addRecipeDetails - gets values from inputs from the form
 - Return - none
 - Functionality - This method posts the recipes from a given form and adds details querying into MongoDB. Sends the data to the '/Addrecipes' api and starts the backend processes.
5. handleRecipeSubmit - handles the stream of actions immediately after clicking 'submit recipes'. Calls postRecipeDetails
 - Param - {array} formDict – stores the input details from the form of adding new recipes
 - Return - none
 - Functionality - This method is a callback function of Form component which sets instance variable of the fields of the form.

This file contains all the routes of the application.

AddRecipeForm.js - ReactJS Class

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

Instance variables :

Ingredients - is a set data structure. No duplicate ingredients are allowed

Dict – Dictionary that stores all of the input details from the form

Functions:

1. handleRecipeSubmit
 - a. Param – event
 - b. Return - none
 - c. Functionality - This method is called by App.js and returns all of the user inputs through the variable dict.
2. constructor
 - a. Param - none
 - b. Return - none
 - c. Functionality - initialize default values
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. render
 - a. Param - none
 - b. Return - none
 - c. Functionality - constructs UI of Add Recipes Form

Form.js - ReactJS Class

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

Instance variables :

Ingredients - is a set data structure. No duplicate ingredients are allowed

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

RecipeList.js - ReactJS functional Component

This component is parent component which calls Recipe component

params: recipe list returned by mongoDB return :
UI Structure of all recipes to be displayed

Recipe.js - ReactJS functional Component

This component structures UI style of a Recipe

params: recipe
return : UI Structure of a recipe

Header.js - ReactJS functional Component

This component structures title of website

params: none
return : UI Structure of a header of website

Login.js - ReactJS function component

This component structures the login form for authentication.

Functions:

1. submitLogin
 - a. Param - on click event of submit button
 - b. Return - User account details
 - c. Functionality - Calls authorize user API and stores the login id in session storage on successful login.

Backend Source files

index.js

This component Index.js , establishes the MongoDB connection.

server.js

This component server.js, connects to the server using the uri mentioned.

.env

This component, .env contains the DB Uri and the port number to connect to the MongoDB

recipesDAO.js

This component, reapiquesDAO contains 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, mail_test_code
- 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. mail_param.js
Stores the authentication password for sending the emails through nodemailer
- 5. postRecipes
 - a. Param - AddRecipeDetails
 - b. Return - res
 - c. Functionality - Adds the values of user inputs of recipes that came from frontend into the MongoDB

recipes.controller.js

This component retrieves the result using the functions in recipesDAO

Functions:

- 1.apiGetRecipes
 - a. Param – req, res, next
 - b. Return - none
 - 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 – none
 - c. Functionality – - retrieves the cuisines response using getRecipes in recipesDAO.
- 3. apiPostRecipes
 - a. Param - req,res,next
 - b. Return - none
 - c. Functionality - calls recipeDAO and passes the req.body and sets the res as json and adds it for posting the recipes. It uses postRecipes from recipesDAO.

recipes.route.js

This component creates a new router object to handle the requests

userauth.route.js

This component creates a new router object to handle the user authorization requests

userauth.service.js

This component retrieves the result using the functions in userAuth Model.

Functions:

1. getAllUsers - Calls model function, gets all the user data and returns it as a json response to front end
2. findUser - Authorizes user by sending request body to model function and returns user data if user credentials are correct, else throws error.
3. saveRecipe - Sends request body to model function to save recipe in the user account data and returns true when updated successfully.

userAuthModel.js

This function connects to the database and performs CRUD operations as requested by service.

Functions:

1. injectDB
 - a. Param - conn
 - b. Return - none
 - c. Functionality - connects to the Mongo database
2. getAllUsers
 - a. Param - none
 - b. Return - list of users
 - c. Functionality - fetches all the users from db
3. findUser

- a. Param - user
 - b. Return - user data
 - c. Functionality - Looks for the user in db
- 4. saveRecipe
 - a. Param - user object containing username and recipeid
 - b. Return - document updated or not
 - c. Functionality - Appends recipe to savedRecipe list of user document