User Profile Management:

Create a new user profile: https://api.flavorlocator.com/users/create

Update user profile: https://api.flavorlocator.com/users/{userID}/update

Retrieve user profile: https://api.flavorlocator.com/users/{userID}

Recipe Management:

Submit a new recipe: https://api.flavorlocator.com/allrecipes/create

Retrieve a recipe: https://api.flavorlocator.com/allrecipes/{recipeID}

Search recipes by ingredients: https://api.flavorlocator.com/allrecipes/search

Save a recipe to the user's collection: https://api.flavorlocator.com/users/{userID}/recipes/save

Review an existing recipe: https://api.flavorlocator.com/recipe/review

Recipe Recommendation/Filtering/Searching:

Get recipe recommendations: https://api.flavorlocator.com/recipes/recommend

Apply dietary restrictions or other filters: https://api.flavorlocator.com/recipes/filter

Ingredient-Based Search:

Search recipes based on ingredients: https://api.flavorlocator.com/recipes/search/ingredients

User Profile Management:

This endpoint will allow a new user to create their own profile with their personal information, which includes items such as: username, email, and even dietary preferences. This will create a new user record in the system and associate it with their unique ID.

Update user profile: Users will use this endpoint to modify their existing profile information, such as updating their display name, adding a profile picture, or changing dietary preferences.

Retrieve user profile: This endpoint enables users to access and view their own profile information. It will allow the users to view the information associated with their profile, such as username, email, dietary preferences, and any other information.

Recipe Management:

Submit a new recipe: Users can add their own recipes to the web application. This will create a new recipe entry associated with the user's ID and include the recipe name, ingredients, any preparation instructions, and nutritional information. It changes the data by adding a new recipe to the database.

Retrieve a recipe: This is the same as the retrieve user profile but in reference to the recipes. Information will not be edited here, just viewed.

Search recipes by ingredients: This endpoint allows users to search for recipes based on ingredients they have or want to use. This serves as a filter dependent on user inputs.

Save a recipe to user's collection: Users also have the option to save recipes to their personal collection for future reference. This action associates a specific recipe with the user's profile, indicating an interest in that recipe.

Recipe Recommendation/Filtering/Searching:

Get recipe recommendations: This endpoint provides recipe suggestions based on the user's preferences and previously saved recipes.

Apply dietary restrictions or allergies filters: This endpoint filters recipe data based on user criteria(inputs), and thus presents recipes that are suitable for their dietary preferences.

Ingredient-Based Search:

Search recipes based on ingredients: Users can search for recipes based on ingredients they have. This endpoint retrieves recipes that match the specified ingredients and shows the results.

These endpoints listed here will allow the users of the application to access and alter their data on the Flavor Locator application, manage their profiles, creating, saving, or searching for recipes, and customizing their recipe recommendations based on their preferences and dietary restrictions.

Example requests/responses

1. User Profile Management:

   - `Create a new user profile`:
   - Request:

     POST /users/create
     ```
     {
        "username": "newuser111",
        "email": "newuser111@yahoo.com",
        "dietaryPreferences": ["vegetarian"]
     }
     ```

   - Successful Response:

     Status: 202 Created
     ```
     {
        "userID": "12345678",
        "message": "User profile has been created successfully."
     }
     ```

   - Error Response (e.g., duplicate email):

     Status: 404 Bad Request
     ```
     {
        "error": "Email is already registered."
     ```

```
        }


- `Update user profile`:

  - Request:


    PUT /users/12345678/update

    {

       "displayName": "Updated User",

       "dietaryPreferences": ["vegan"]

    }


  - Successful Response:


    Status: 200 OK

    {

       "message": "User profile has been updated successfully."

    }


  - Error Response (e.g., unauthorized):


    Status: 401 Unauthorized

    {

       "error": "Unauthorized: User does not have permission to make changes this profile."

    }

  -Retrieve user profile`:

  - Request:


    GET /users/12345678
```

- Successful Response:

  Status: 200 OK

  ```json
  {
     "userID": "12345678",
     "username": "newuser111",
     "email": "newuser111@yahoo.com",
     "dietaryPreferences": ["vegan"],
     "displayName": "Updated User"
  }
  ```

2. Recipe Management:

  - `Submit a new recipe`:
    - Request:

    POST /recipes/create
    ```json
    {
       "recipeName": "Texas Chili",
       "ingredients": ["Beef", "Chili Powder", "Pinto Beans"],
       "instructions": ".."
    }
    ```

  - Successful Response:

    Status: 201 Created

```
{
    "recipeID": "1987654",
    "message": "Recipe has been created successfully."
}
```

- Error Response (e.g., missing recipe ingredients):

```
Status: 404 Bad Request
{
    "error": "Recipe ingredients are required for submission."
}
```

- `Retrieve a recipe`:
  - Request:

```
GET /recipes/1987654
```

  - Successful Response:

```
Status: 200 OK
{
    "recipeName": "Texas Chili",
    "ingredients": ["Beef", "Chili Powder", "Pinto Beans"],
    "instructions": "...",
    "nutritionInfo": { ... }
}
```

3. **Recipe Recommendation and Filtering**:

  - `Get recipe recommendations`:
   - Request:

     GET /recipes/recommend?userID=123456

   - Successful Response:

     Status: 200 OK
     {
        "recommendedRecipes": [...]
     }

  - `Apply dietary restrictions or allergies filters`:
   - Request:

     GET /recipes/filter?userID=123456&dietaryRestrictions=vegan&allergies=peanut

   - Successful Response:

     Status: 200 OK
     {
        "filteredRecipes": [...]
     }

4. **Ingredient-Based Search**:

  - `Search recipes based on ingredients`:
   - Request:

     GET /recipes/search/ingredients?ingredients=chicken,pepper,broccoli,milk

   - Successful Response:

     Status: 200 OK
     {
        "matchingRecipes": [...]
     }

5. **Nutritional Information**:

  - `Retrieve nutritional information for a recipe`:
   - Request:

     GET /recipes/1987654/nutrition

   - Successful Response:

     Status: 200 OK
     {
        "nutritionInfo": { ... }

}

- Error Response (e.g., invalid recipe ID):

Status: 404 Not Found
{
    "error": "Recipe could not be located."
}