**Documentation of the Recipe API**

**1. Necessary modules used:**

- express: Used to create the web server and handles HTTP requests.
- fs: Interacts with the file system to read and write the recipes.json file.
- path: Constructs file paths in a platform-independent way.
- uuid: Generates unique identifiers for recipes (but note that it's imported but not used consistently).

**2. Setting up the Express app:**

- Create an Express app instance.
- Use app.use(express.json()) to parse incoming JSON data in requests.

**3. Defining file paths:**

- Set dataFolderPath to the 'data' directory.
- Set recipesFilePath to the 'recipes.json' file within the data folder.

**4. Authentication function:**

- authenticateUser checks for valid username and password (currently hardcoded to 'admin' and 'admin').

**5. Recipe ID generation:**

- generateRecipeId increments a counter to create unique recipe IDs.

**6. Adding a new recipe:**

- addNewRecipe assigns a new ID, appends the recipe to the data, and writes the updated recipes to the JSON file.

**7. User validation middleware:**

- validateUser checks for valid credentials using authenticateUser and sends error responses if authentication fails.

**8. API routes:**

- **GET /:** Returns a basic message indicating the API is running.
- **GET /api/recipes/:category:** Retrieves recipes by category.
- **GET /api/recipes:** Retrieves all recipes.
- **GET /api/recipes/name/:recipeName:** Retrieves a recipe by name.
- **GET /api/recipes/id/:recipeId:** Retrieves a recipe by ID.

- **DELETE /api/recipes/:recipeId:** Deletes a recipe by ID.
- **POST /api/recipes/addNewRecipe:** Adds a new recipe (note the inconsistency with other POST routes).
- **POST /api/recipes:** Adds a new recipe (using a more consistent approach with other routes).
- **PUT /api/recipes/:recipeId:** Updates an existing recipe.
- **DELETE /api/recipes/:recipeId:** Deletes a recipe by ID (duplicate of a previous route).

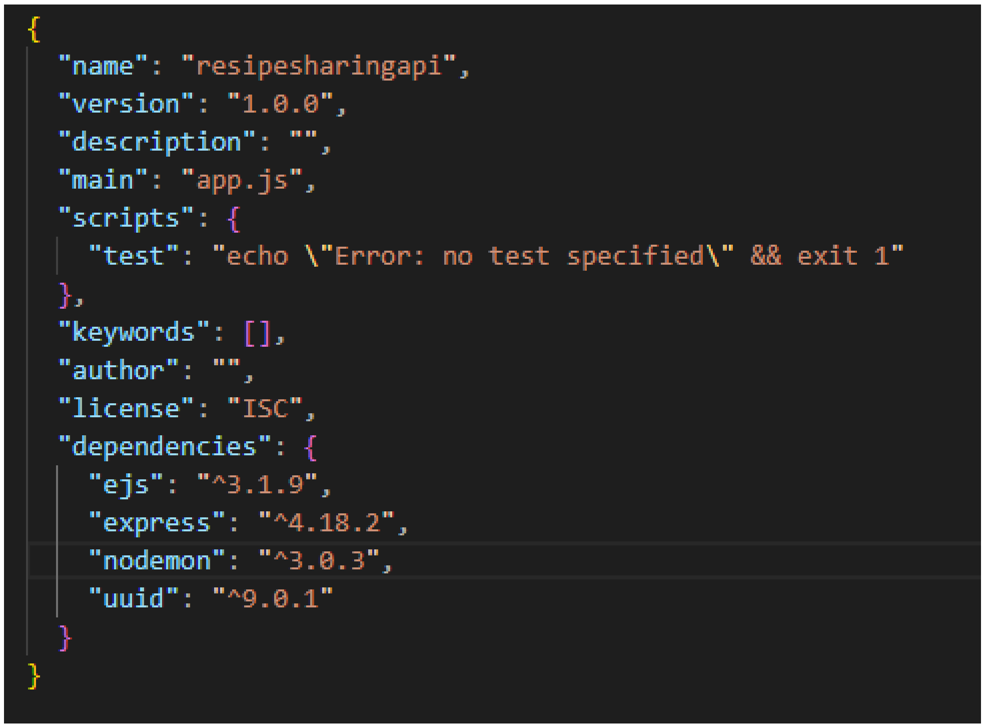
**9. Initial data population:**

- Reads the recipes.json file and adds a sample recipe if the file is empty (this code seems unnecessary for typical usage).

**10. Starting the server:**

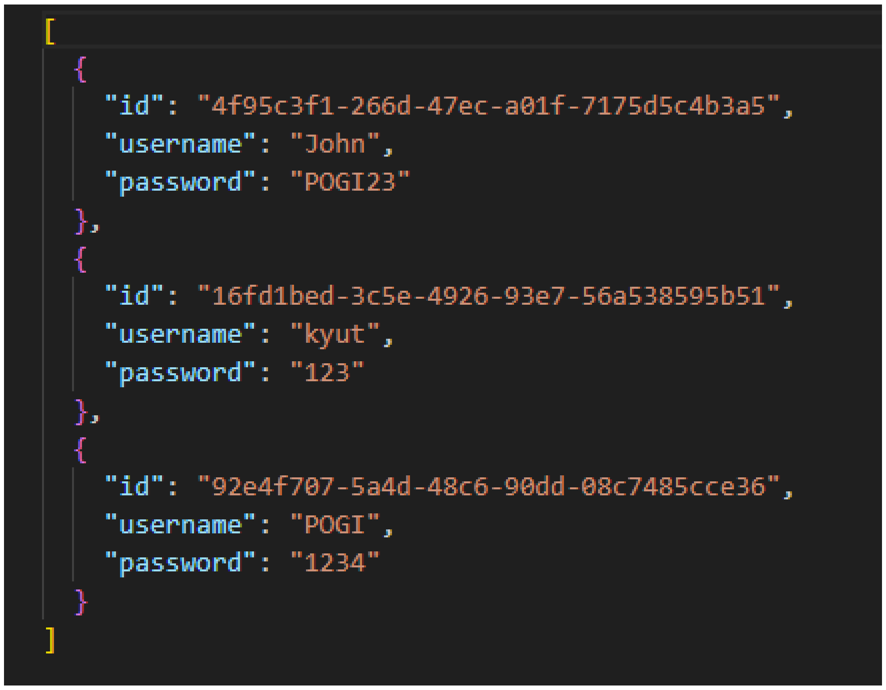
- Listens on the specified port (defaulting to 3001) and logs a message indicating the server is running.

**.json files created**



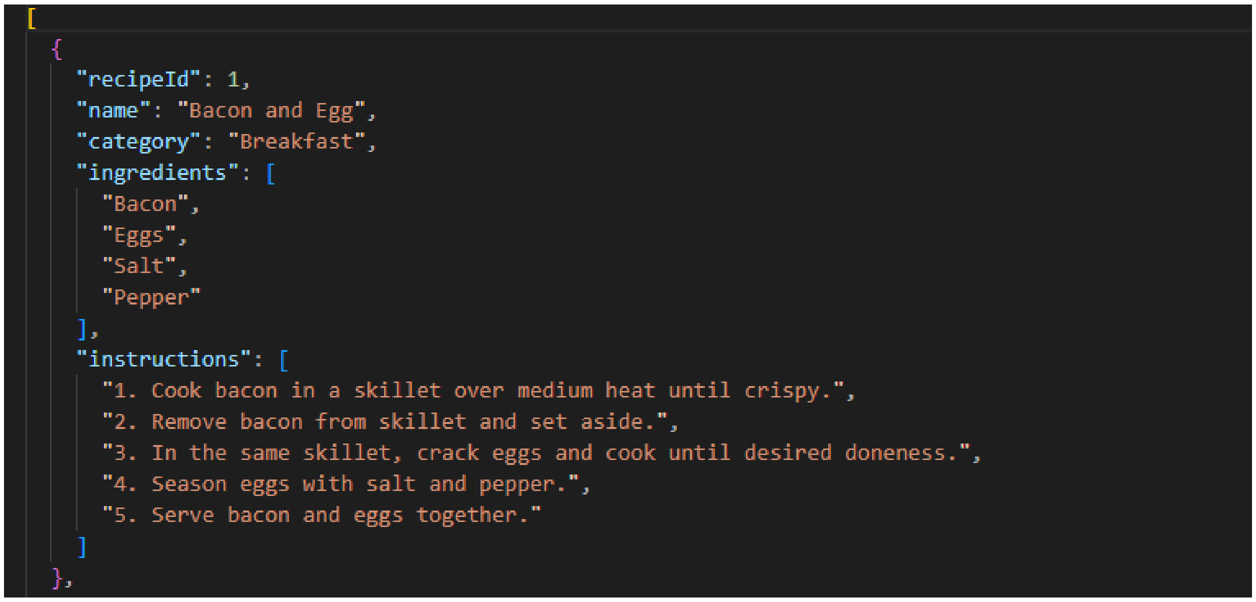
```json
{
  "name": "resipesharingapi",
  "version": "1.0.0",
  "description": "",
  "main": "app.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "ejs": "^3.1.9",
    "express": "^4.18.2",
    "nodemon": "^3.0.3",
    "uuid": "^9.0.1"
  }
}
```

**package.json –** this describes the project by its version, dependencies set for the recipe-sharing API, and the web framework used which is Express.



```json
[
  {
    "id": "4f95c3f1-266d-47ec-a01f-7175d5c4b3a5",
    "username": "John",
    "password": "POGI23"
  },
  {
    "id": "16fd1bed-3c5e-4926-93e7-56a538595b51",
    "username": "kyut",
    "password": "123"
  },
  {
    "id": "92e4f707-5a4d-48c6-90dd-08c7485cce36",
    "username": "POGI",
    "password": "1234"
  }
]
```

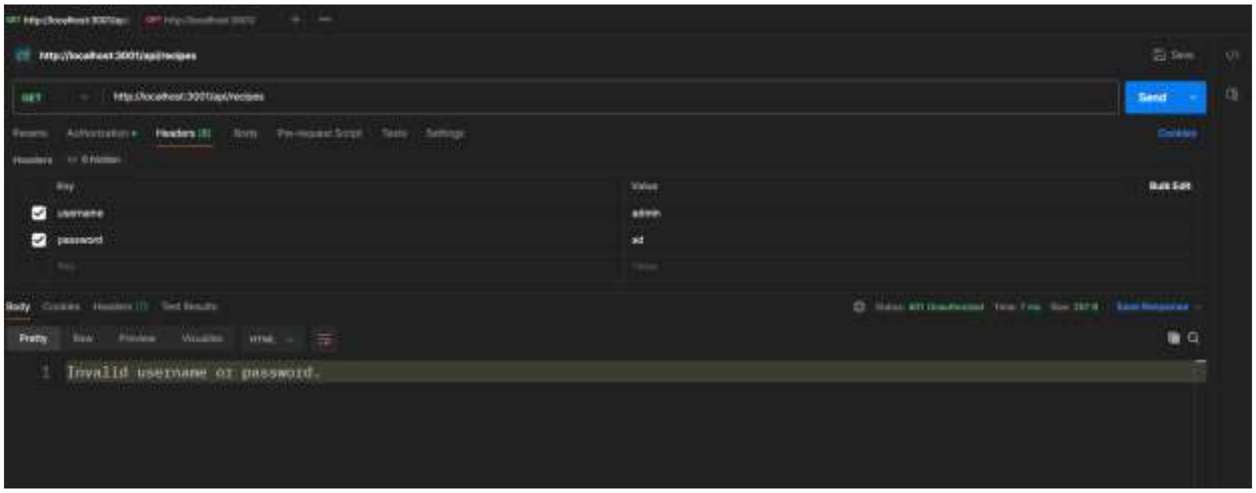
**users.json** - Where the id, username, and password set by the users are stored.



```json
[
  {
    "recipeId": 1,
    "name": "Bacon and Egg",
    "category": "Breakfast",
    "ingredients": [
      "Bacon",
      "Eggs",
      "Salt",
      "Pepper"
    ],
    "instructions": [
      "1. Cook bacon in a skillet over medium heat until crispy.",
      "2. Remove bacon from skillet and set aside.",
      "3. In the same skillet, crack eggs and cook until desired doneness.",
      "4. Season eggs with salt and pepper.",
      "5. Serve bacon and eggs together."
    ]
  },
```

**recipes.json –** This is where the particular recipes to be accessed by the user are stored.

## Snippets of the Working API


http://localhost:3001/api/recipes

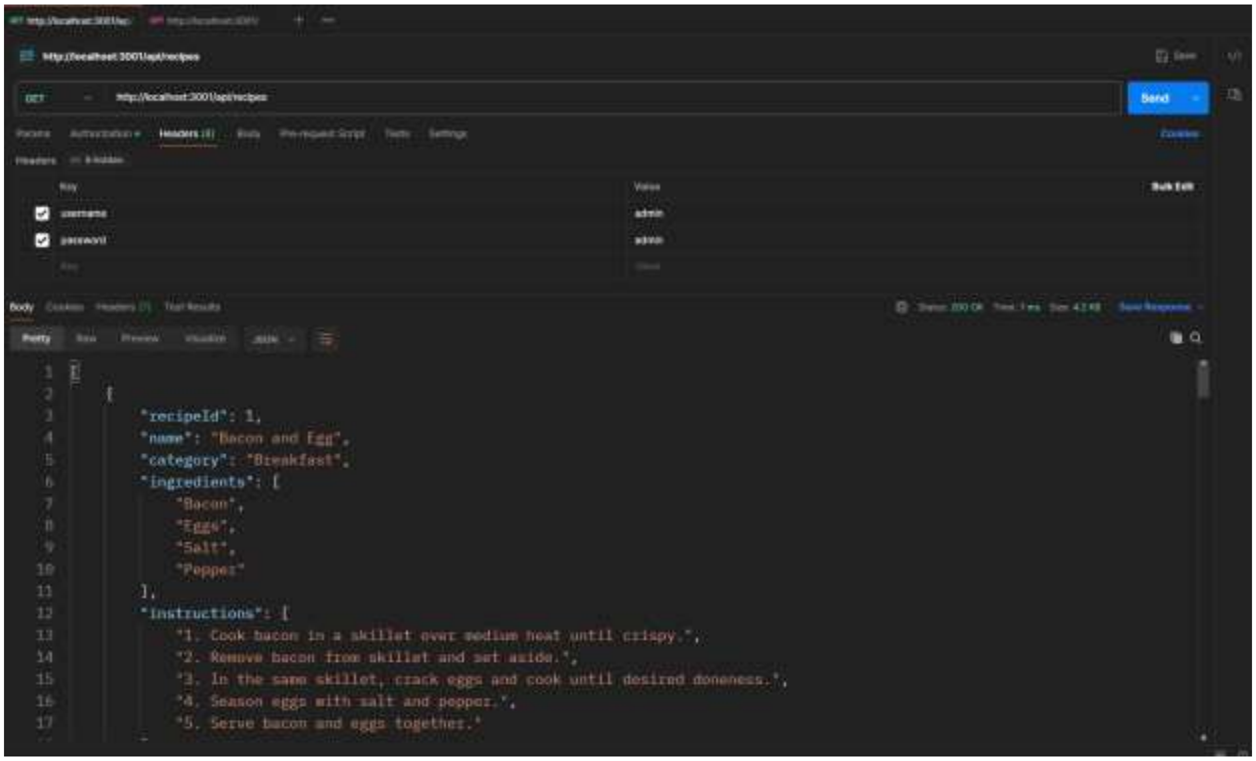
GET http://localhost:3001/api/recipes

| Key | Value |
|---|---|
| username | admin |
| password | ad |

Status: 401 Unauthorized

```
Invalid username or password.
```

If the username and password don’t match


http://localhost:3001/api/recipes

GET http://localhost:3001/api/recipes

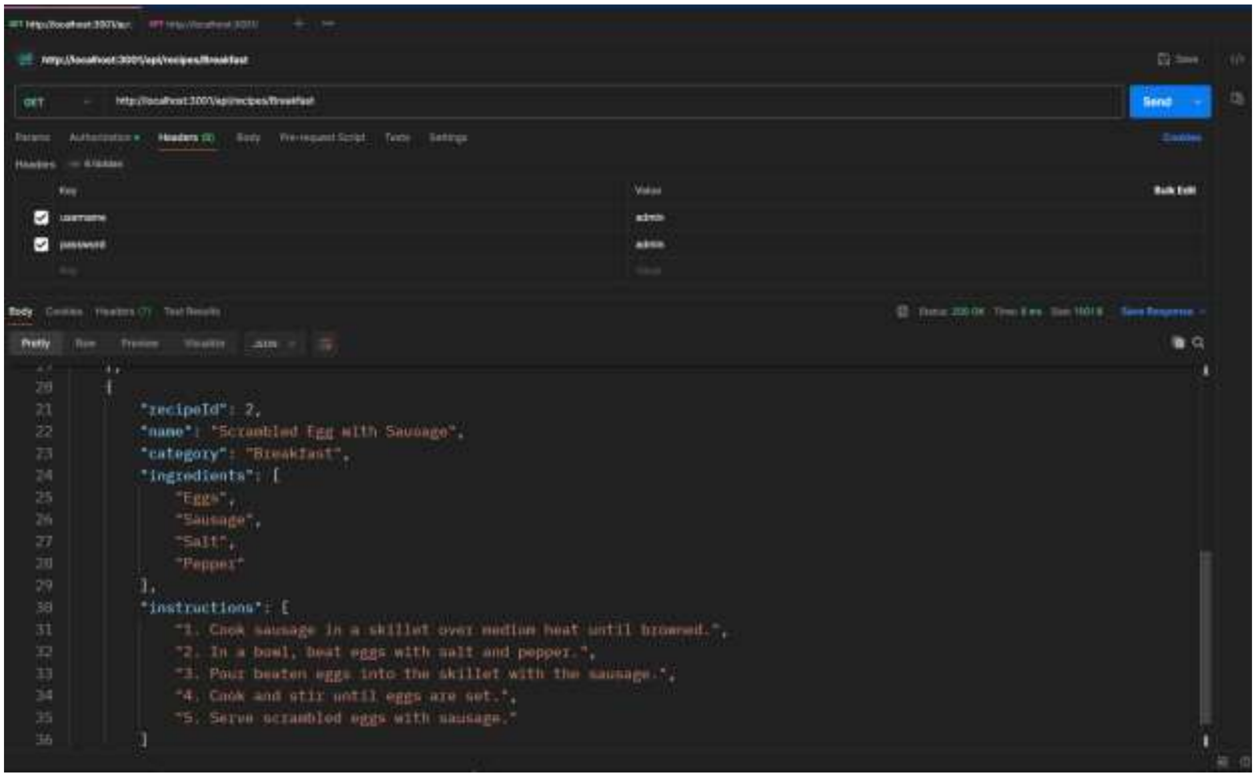
| Key | Value |
|---|---|
| username | admin |
| password | admin |

Status: 200 OK

```
[
    {
        "recipeId": 1,
        "name": "Bacon and Egg",
        "category": "Breakfast",
        "ingredients": [
            "Bacon",
            "Eggs",
            "Salt",
            "Pepper"
        ],
        "instructions": [
            "1. Cook bacon in a skillet over medium heat until crispy.",
            "2. Remove bacon from skillet and set aside.",
            "3. In the same skillet, crack eggs and cook until desired doneness.",
            "4. Season eggs with salt and pepper.",
            "5. Serve bacon and eggs together."
```

**GET** http://localhost:3001/api/recipes

To access all the recipe



http://localhost:3001/api/recipes/Breakfast

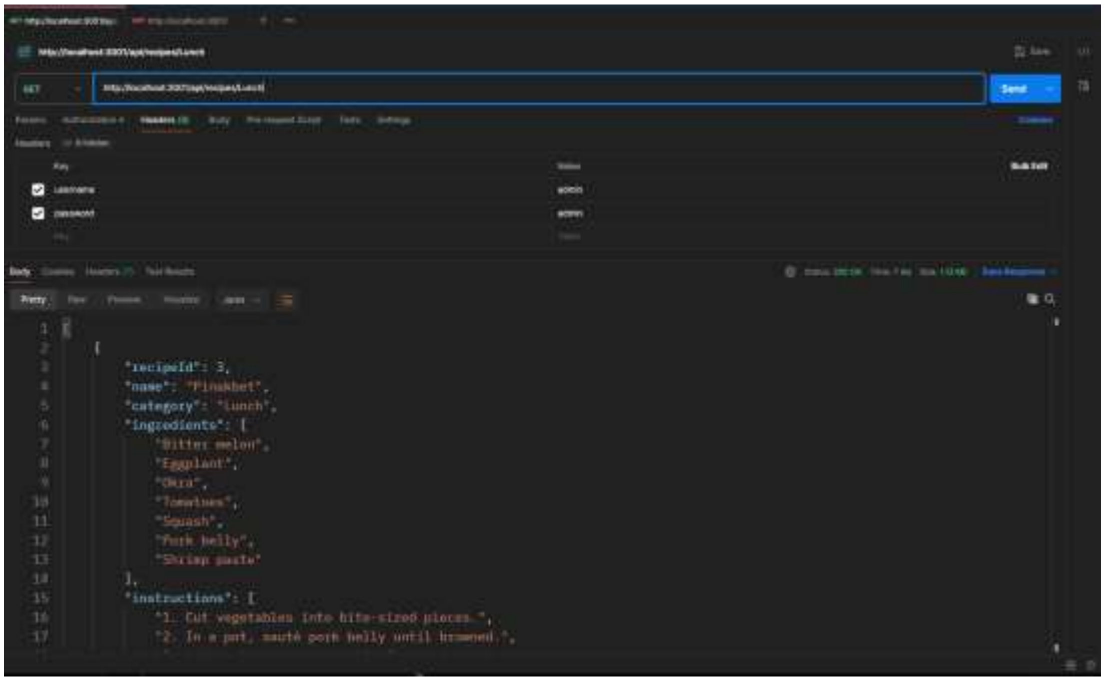
GET http://localhost:3001/api/recipes/Breakfast

| Key | Value |
|---|---|
| ✓ username | admin |
| ✓ password | admin |

```
    },
    {
        "recipeId": 2,
        "name": "Scrambled Egg with Sausage",
        "category": "Breakfast",
        "ingredients": [
            "Eggs",
            "Sausage",
            "Salt",
            "Pepper"
        ],
        "instructions": [
            "1. Cook sausage in a skillet over medium heat until browned.",
            "2. In a bowl, beat eggs with salt and pepper.",
            "3. Pour beaten eggs into the skillet with the sausage.",
            "4. Cook and stir until eggs are set.",
            "5. Serve scrambled eggs with sausage."
        ]
```

**GET** http://localhost:3001/api/recipes/Breakfast

To access all the recipe of breakfast category



http://localhost:3001/api/recipes/Lunch

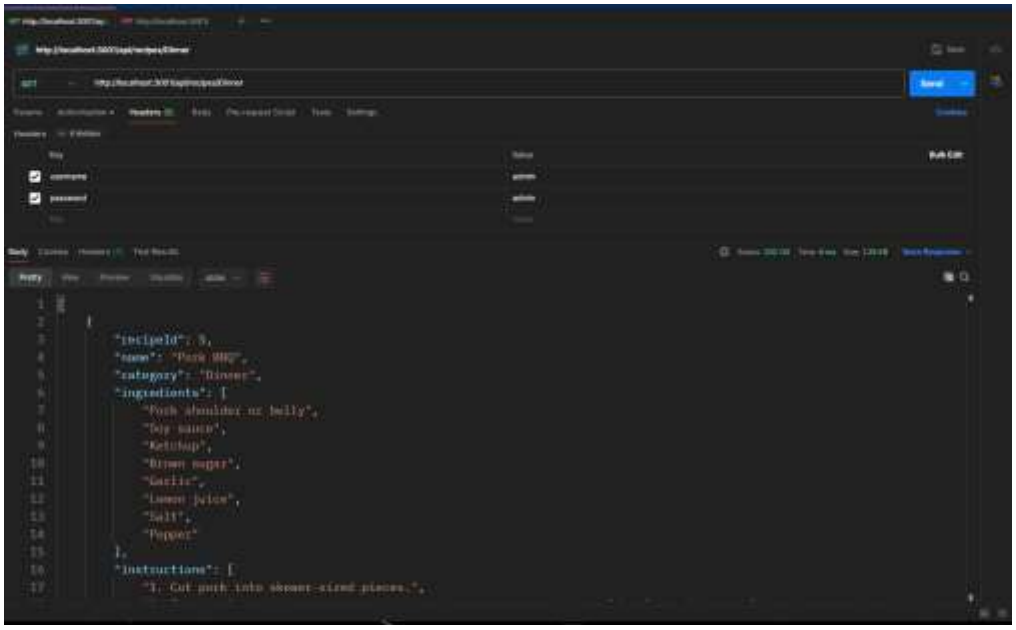
GET http://localhost:3001/api/recipes/Lunch

| Key | Value |
|---|---|
| ✓ username | admin |
| ✓ password | admin |

```
[
    {
        "recipeId": 3,
        "name": "Pinakbet",
        "category": "Lunch",
        "ingredients": [
            "Bitter melon",
            "Eggplant",
            "Okra",
            "Tomatoes",
            "Squash",
            "Pork belly",
            "Shrimp paste"
        ],
        "instructions": [
            "1. Cut vegetables into bite-sized pieces.",
            "2. In a pot, sauté pork belly until browned.",
```

**GET** http://localhost:3001/api/recipes/Lunch

To access all the recipe in the lunch category



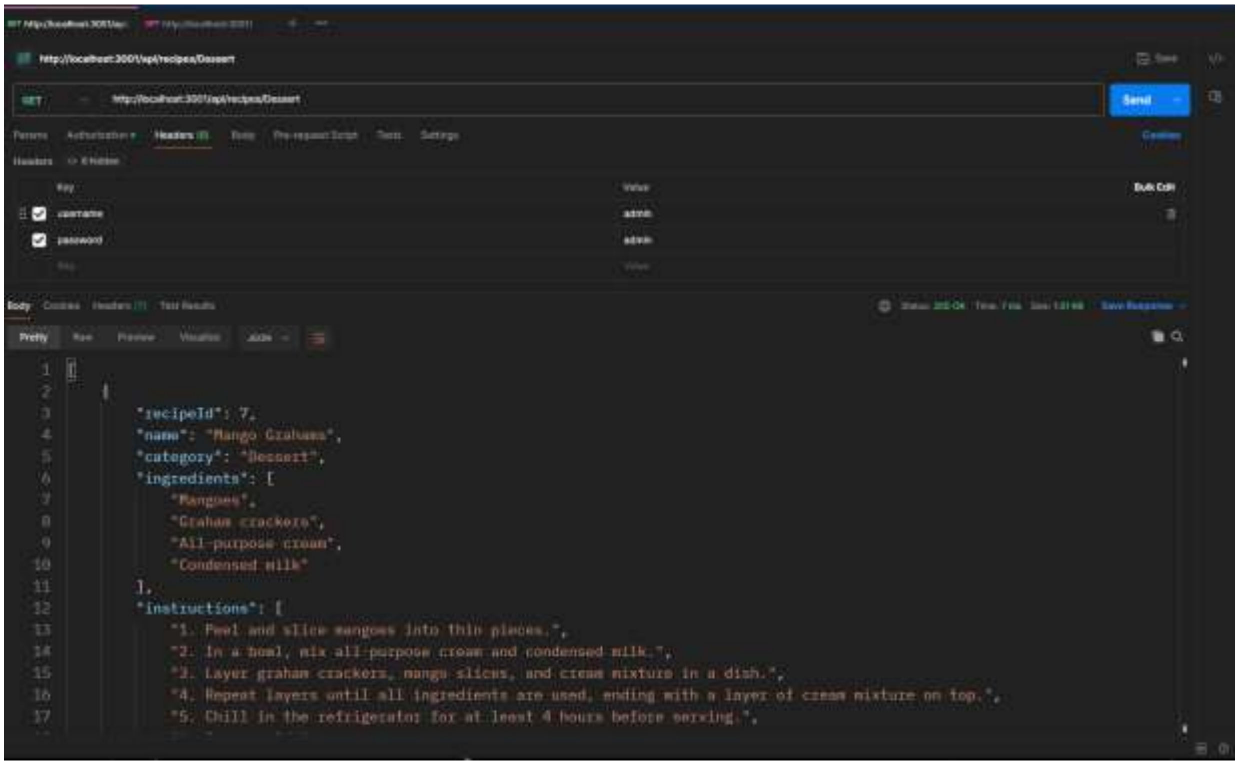
GET http://localhost:3001/api/recipes/Dinner

| Key | Value |
|---|---|
| username | admin |
| password | admin |

```
[
    {
        "recipeId": 5,
        "name": "Pork BBQ",
        "category": "Dinner",
        "ingredients": [
            "Pork shoulder or belly",
            "Soy sauce",
            "Ketchup",
            "Brown sugar",
            "Garlic",
            "Lemon juice",
            "Salt",
            "Pepper"
        ],
        "instructions": [
            "1. Cut pork into skewer-sized pieces.",
```

**GET** http://localhost:3001/api/recipes/Dinner

To access all the recipe in the dinner category



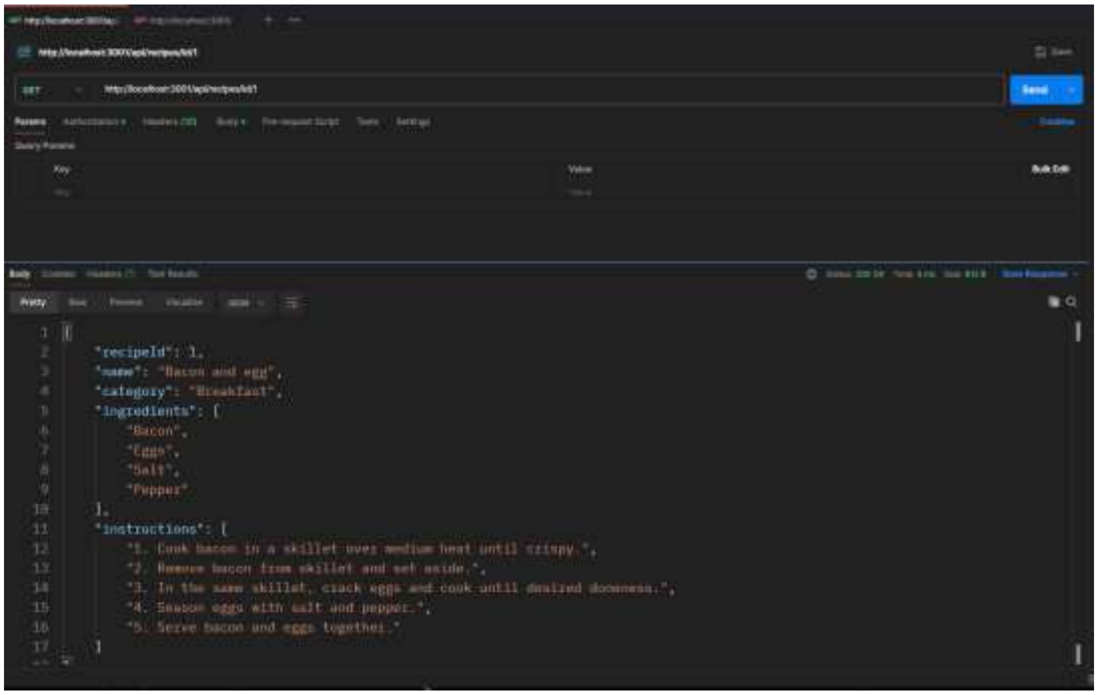
GET http://localhost:3001/api/recipes/Dessert

| Key | Value |
|---|---|
| username | admin |
| password | admin |

```
[
    {
        "recipeId": 7,
        "name": "Mango Grahams",
        "category": "Dessert",
        "ingredients": [
            "Mangoes",
            "Graham crackers",
            "All-purpose cream",
            "Condensed milk"
        ],
        "instructions": [
            "1. Peel and slice mangoes into thin pieces.",
            "2. In a bowl, mix all-purpose cream and condensed milk.",
            "3. Layer graham crackers, mango slices, and cream mixture in a dish.",
            "4. Repeat layers until all ingredients are used, ending with a layer of cream mixture on top.",
            "5. Chill in the refrigerator for at least 4 hours before serving.",
```

**GET** http://localhost:3001/api/recipes/Dessert

To access all the recipe in the dessert category

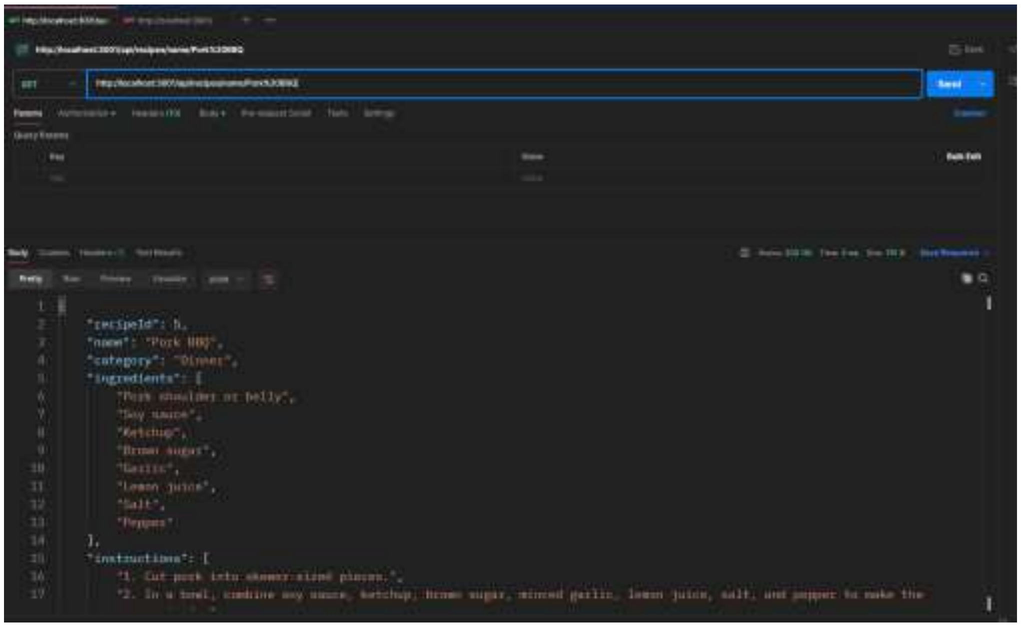


```
GET    http://localhost:3001/api/recipes/id/1
```

```
{
    "recipeId": 1,
    "name": "Bacon and egg",
    "category": "Breakfast",
    "ingredients": [
        "Bacon",
        "Eggs",
        "Salt",
        "Pepper"
    ],
    "instructions": [
        "1. Cook bacon in a skillet over medium heat until crispy.",
        "2. Remove bacon from skillet and set aside.",
        "3. In the same skillet, crack eggs and cook until desired doneness.",
        "4. Season eggs with salt and pepper.",
        "5. Serve bacon and eggs together."
    ]
```

**GET** http://localhost:3001/api/recipes/id/:recipeId

To access recipe using recipe ID

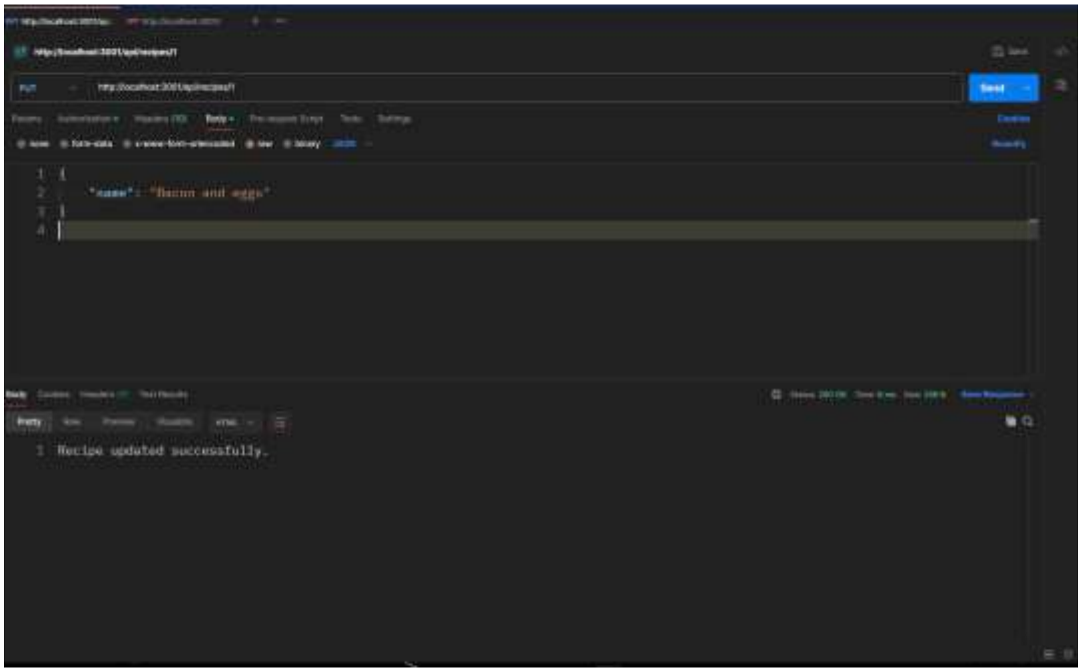


```
GET    http://localhost:3001/api/recipes/name/Pork%20BBQ
```

```
{
    "recipeId": 5,
    "name": "Pork BBQ",
    "category": "Dinner",
    "ingredients": [
        "Pork shoulder or belly",
        "Soy sauce",
        "Ketchup",
        "Brown sugar",
        "Garlic",
        "Lemon juice",
        "Salt",
        "Pepper"
    ],
    "instructions": [
        "1. Cut pork into skewer-sized pieces.",
        "2. In a bowl, combine soy sauce, ketchup, brown sugar, minced garlic, lemon juice, salt, and pepper to make the
```

**GET** http://localhost:3001/api/recipes/name/:name

To access recipe using name of the recipe



PUT http://localhost:3001/api/recipes/1

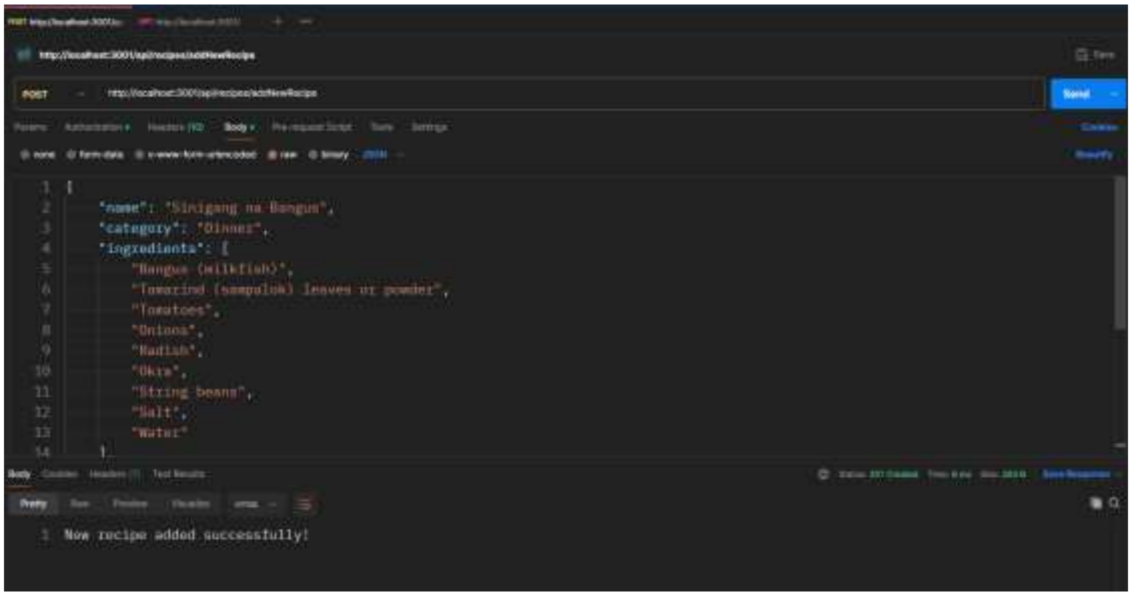
```
{
    "name": "Bacon and eggs"
}

```

```
Recipe updated successfully.
```

**PUT** http://localhost:3001/api/recipes/:recipeID

To edit the existing recipe.



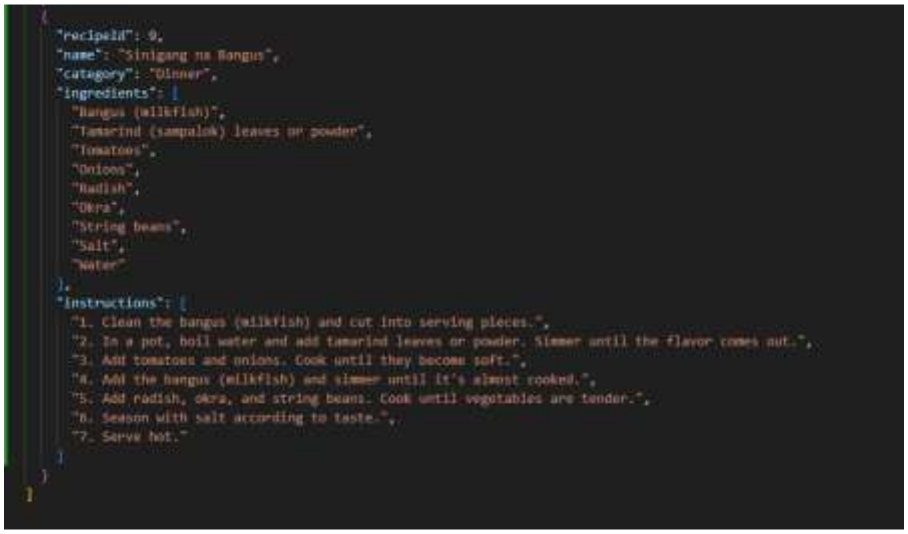
POST http://localhost:3001/api/recipes/addNewRecipe

```
{
    "name": "Sinigang na Bangus",
    "category": "Dinner",
    "ingredients": [
        "Bangus (milkfish)",
        "Tamarind (sampalok) leaves or powder",
        "Tomatoes",
        "Onions",
        "Radish",
        "Okra",
        "String beans",
        "Salt",
        "Water"
    ],
```

```
New recipe added successfully!
```

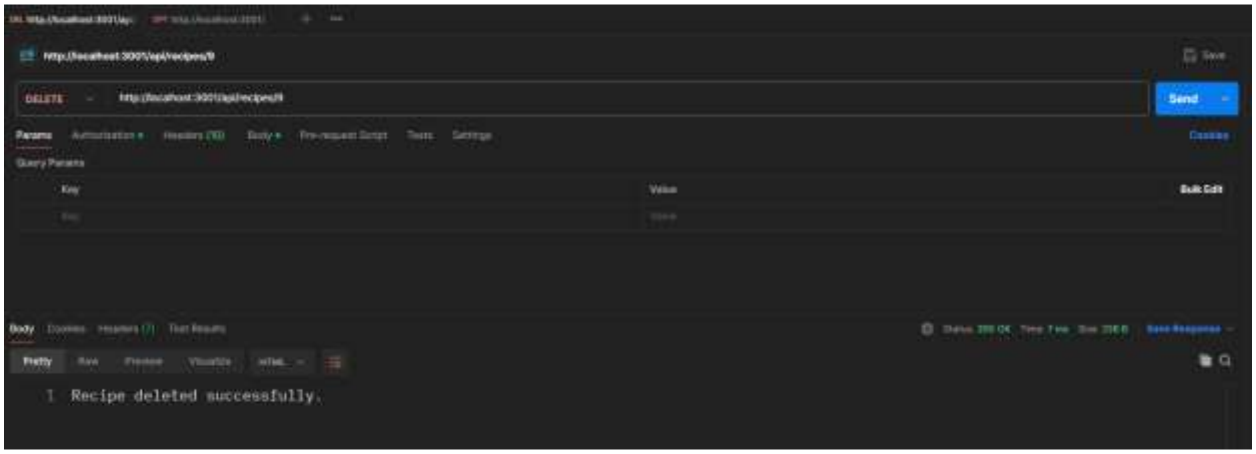
POST http://localhost:3001/api/recipes/addNewRecipe

Add new recipe. When sending POST request, make sure to include all the necessary data



```
  {
    "recipeId": 9,
    "name": "Sinigang na Bangus",
    "category": "Dinner",
    "ingredients": [
      "Bangus (milkfish)",
      "Tamarind (sampalok) leaves or powder",
      "Tomatoes",
      "Onions",
      "Radish",
      "Okra",
      "String beans",
      "Salt",
      "Water"
    ],
    "instructions": [
      "1. Clean the bangus (milkfish) and cut into serving pieces.",
      "2. In a pot, boil water and add tamarind leaves or powder. Simmer until the flavor comes out.",
      "3. Add tomatoes and onions. Cook until they become soft.",
      "4. Add the bangus (milkfish) and simmer until it's almost cooked.",
      "5. Add radish, okra, and string beans. Cook until vegetables are tender.",
      "6. Season with salt according to taste.",
      "7. Serve hot."
    ]
  }
]
```

It will automatically be added to recipe.json file and it will automatically generate recipeId to avoid same ID.

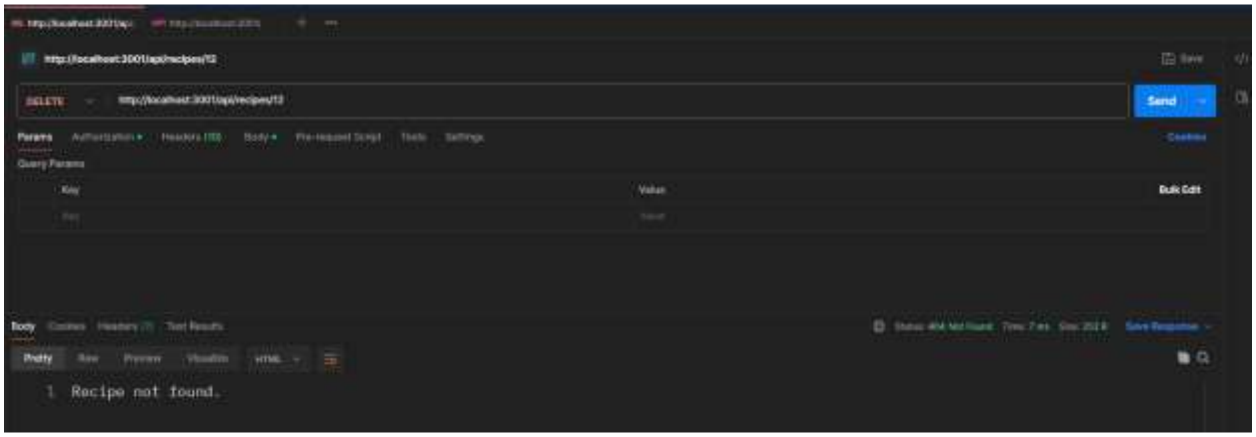


```
DELETE   http://localhost:3001/api/recipes/9
```

```
1  Recipe deleted successfully.
```

DELETE http://localhost:3001/api/recipes/:recipeId

To delete recipe

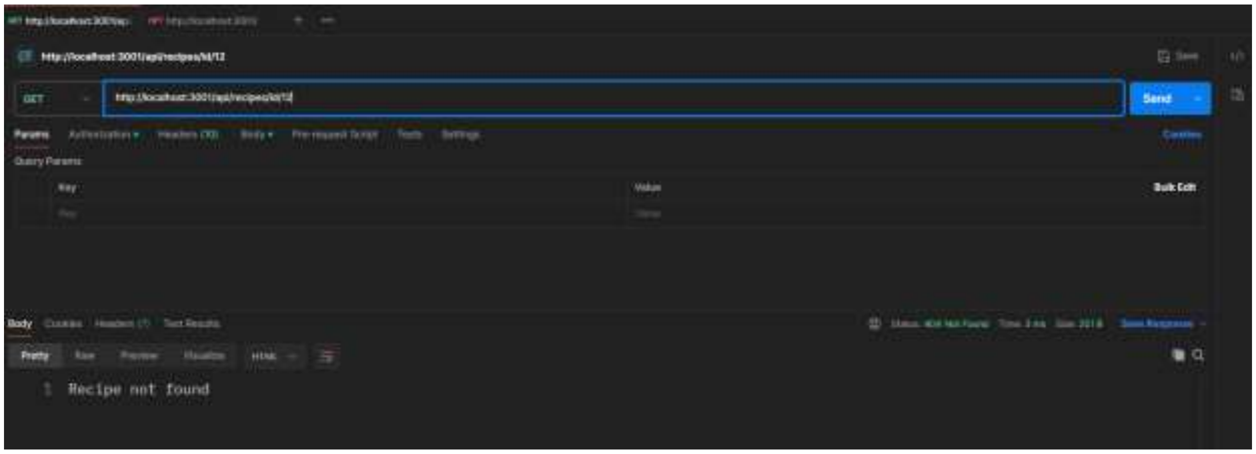


DELETE http://localhost:3001/api/recipes/12

Status: 404 Not Found

```
Recipe not found.
```

If the recipe you want to delete does not exist.

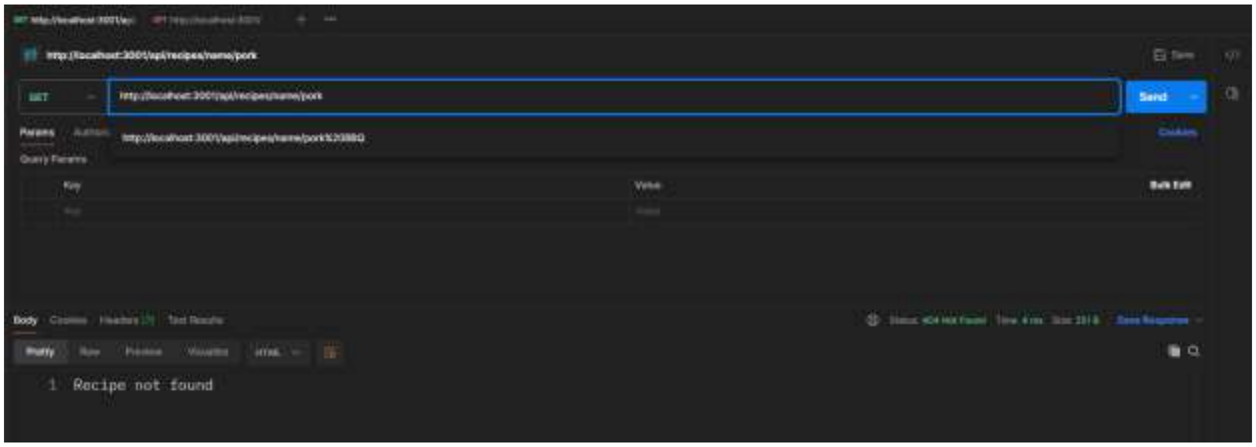


GET http://localhost:3001/api/recipes/id/12

Status: 404 Not Found

```
Recipe not found
```

If the recipe you want to get using id does not exist

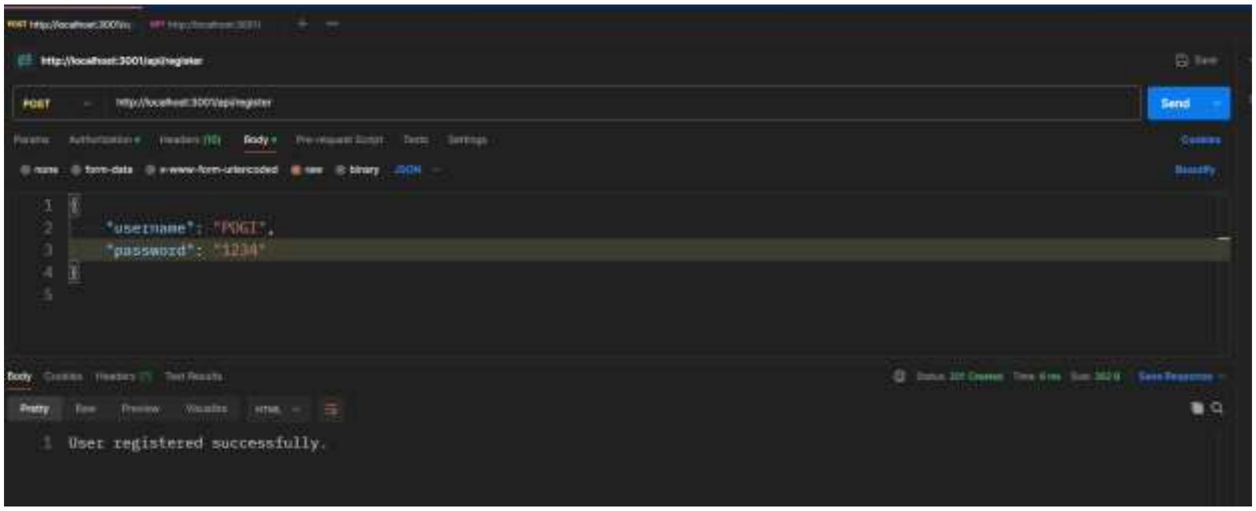


GET http://localhost:3001/api/recipes/name/pork

Status: 404 Not Found

```
Recipe not found
```

If the recipe you want to get using name does not exist



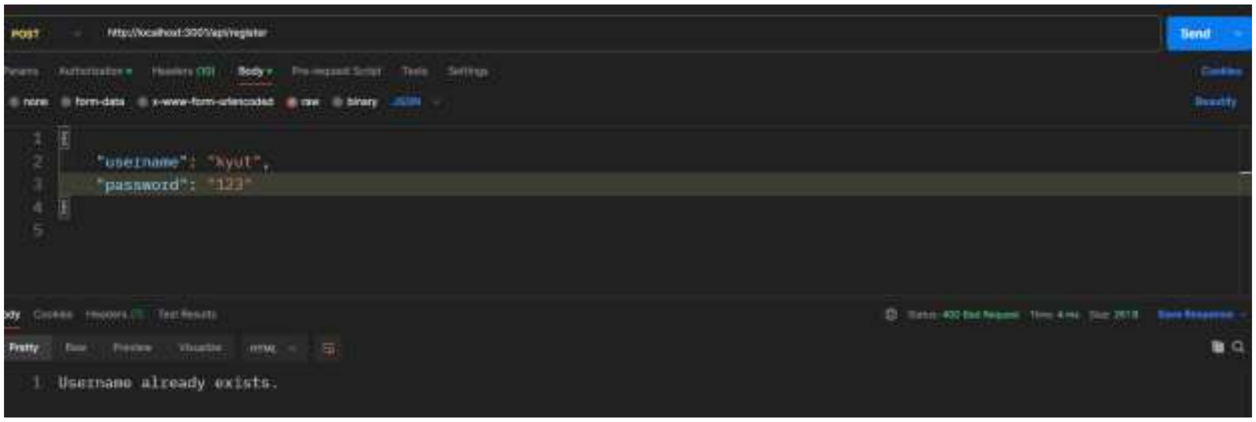
POST http://localhost:3001/api/register

```
{
    "username": "POGI",
    "password": "1234"
}
```

Status: 201 Created Time: 6 ms Size: 262 B

```
User registered successfully.
```

If the user successfully registered an account



POST http://localhost:3001/api/register

```
{
    "username": "kyut",
    "password": "123"
}
```

Status: 400 Bad Request Time: 4 ms Size: 261 B

```
Username already exists.
```

Dialogue shown when the user is setting a new username that already exists in the system.