Technical Documentation of Meal DB

1. Code Guidelines

1.1. Naming

- Variables: camelCase (e.g., mealData).
- Files: Lowercase with hyphens (e.g., meal-details.js).
- Folders: Group by function (e.g., /utils/reaction).

1.2. Code Style

- Use semicolons, modern JavaScript features, and handle errors with try-catch.

2. Project Overview & Function Descriptions

This project retrieves meal details and interacts with Discord:

- getMealFromDB(category):
  - Fetches a random meal based on a given category.
  - If the category starts with "filter:", it returns just the meal names.
  - Otherwise, it returns the meal's name, recipe (instructions), and ingredients.
  - For no category, it fetches a random meal and returns its recipe and ingredients.
- getMeal(req, res, next):
  - Grabs the 'category' from the incoming request.
  - Fetches the meal data based on this category.
  - Sends this meal data to Discord using postRequestToDiscord.
- Utility Functions:
  - extractMealName(mealData): Retrieves the name of a single meal.
  - extractMealNames(data): Gets the names of multiple meals and formats them in a list.
  - extractIngredients(meal): Processes and lists out the ingredients used in a meal.

3. Technologies

- Express.js: For web request management.
- Database/API: Interacts with a presumed database or service to get meal details.
- Discord Integration: Shares meal details on Discord.

4. Description of the Reaction

This reaction in a 3 in 1 reaction:

- Random meal
  Using a call, you get a random meal, its recipe and its ingredient.

- Targeted meal
  Using a call, you get the meal you imputed, it’s recipe and ingredient.
- Targeted ingredient
  Usign a call, you get a list of meal that all include the ingredient you listed.