Recipes

API Project

Name: **Messaoud Ben ALi Haddad**
Major: IT
Module: IT 325

Contents

1 INTRODUCTION **4**

2 UTILITY **5**

3 Token Generation Endpoint **6**
- 3.1 Endpoint Summary . . . 6
- 3.2 OAuth 2.0 Grant Type . . . 6
- 3.3 Request Parameters . . . 6
- 3.4 Example Response . . . 7
- 3.5 Security Considerations . . . 7
- 3.6 Conclusion . . . 7

4 RecipeData Schema **8**
- 4.1 Schema Definition . . . 8
- 4.2 Usage in API Endpoints . . . 8
- 4.3 Conclusion . . . 9

5 GET RECIPE Endpoint **9**
- 5.1 Endpoint Summary . . . 9
- 5.2 Functionality . . . 9
- 5.3 Request Parameters . . . 9
- 5.4 Response Model . . . 9
- 5.5 External API Integration . . . 9
- 5.6 Conclusion . . . 10

6 GET RECIPES Endpoint **10**
- 6.1 Endpoint Summary . . . 10
- 6.2 Functionality . . . 10
- 6.3 Request Parameters . . . 10
- 6.4 Response Model . . . 10
- 6.5 Response Description . . . 10
- 6.6 Error Handling . . . 10
- 6.7 Conclusion . . . 11

7 CREATE RECIPE Endpoint **11**
- 7.1 Endpoint Summary . . . 11
- 7.2 Functionality . . . 11
- 7.3 Request Parameters . . . 11
- 7.4 Response Model . . . 11
- 7.5 Response Description . . . 11
- 7.6 Error Handling . . . 12
- 7.7 Conclusion . . . 12

8 Update Recipe Endpoint **12**
- 8.1 Endpoint Summary . . . 12
- 8.2 Functionality . . . 12
- 8.3 Request Parameters . . . 12
- 8.4 Response Model . . . 12
- 8.5 Response Description . . . 13
- 8.6 Error Handling . . . 13
- 8.7 Conclusion . . . 13

9 DELETE recipe Endpoint **13**
9.1 Endpoint Summary . . . 13
9.2 Functionality . . . 13
9.3 Request Parameters . . . 13
9.4 Response Model . . . 13
9.5 Response Description . . . 14
9.6 Error Handling . . . 14
9.7 Debugging Information . . . 14
9.8 Conclusion . . . 14
9.9 Requirements.txt . . . 14
9.9.1 Requirements Content . . . 14
9.9.2 Explanation . . . 14
9.9.3 Conclusion . . . 14

1 INTRODUCTION

The growing interest in a variety of culinary experiences has caused the investigation and recording of different cuisine recipes to receive more attention. The creation and distribution of culinary marvels has gained international recognition in recent years. The global exchange of culinary concepts is booming, and one of the most important aspects of this cultural exchange is the recording of recipes. In this setting, experimenting with food recipes turns into a voyage of discovery that brings individuals together via common culinary interests. Moving forward, we must recognise that food is important in ways more than just sustenance. Food is a joy, a cultural ambassador, and a vehicle for people to express their individuality and creativity. This in-depth examination of culinary recipes will cover the history of culinary customs, the changing face of international cuisine, and a forward-thinking viewpoint that celebrates the dynamic and varied nature of the culinary industry. Through shared meals and conversations, we take you on a culinary journey that transcends the kitchen and hits the core of human connection as we explore the complexities of ingredients, techniques, and cultural influences.

Background

The fascinating realm of culinary recipes is a dynamic and always changing tapestry that deftly combines the many cultures, age-old customs, and personal preferences of individuals from all over the world. The development, recording, and dissemination of food recipes have become significant manifestations of our cultural history of cooking, going beyond simple lists of ingredients and directions. These gastronomic creations function as a captivating portal, providing insights into the diverse tastes and culinary customs that characterise various societies and areas. Recipe sharing and recording transcends the kitchen and becomes an essential part of our cultural heritage, a complex ballet of flavours and skills. As societies become increasingly networked, there is a greater flow of culinary ideas. A culinary symphony is created by individuals, seasoned chefs, and enthusiastic food aficionados who are constantly experimenting and exploring a wide range of ingredients, cutting-edge methods, and varied cultural influences. The changing nature of the food industry necessitates both an understanding of its rich past and a forward-thinking mindset. Thorough knowledge and careful record-keeping of food recipes are now vital foundations for maintaining long-standing culinary customs, encouraging creativity, and adjusting to the changing nutritional requirements of an international community. This complex interaction between invention and tradition serves as the foundation for a gastronomic adventure that honours variety, originality, and the shared delight of creating and indulging in mouthwatering meals.

2 UTILITY

The Recipe API project intends to provide a comprehensive and user-friendly interface for obtaining and managing food-related data. This technical study describes the API's objectives, methodology, important features, and prospective use cases.

Objectives

The primary objectives of Recipes API project are as follows:

- To centralize and standardize access to food recipes data from various sources.
- To offer a RESTful API that enables users to retrieve real-time and historical recipe data.
- To provide endpoints for submitting new recipe data and updating existing records.
- To implement authentication mechanisms to secure access to sensitive data.
- To support integration with external applications, allowing developers to incorporate emission data into their projects.

Methodology

The development of the Recipes API follows an iterative and collaborative approach. The project involves the following key steps:

1. Requirement Analysis: Identifying the specific needs of users and stakeholders.
2. API Design: Defining the structure of API endpoints, data models, and authentication mechanisms.
3. Implementation: Developing the API using a suitable web framework (e.g., FastAPI, Django).
4. Testing: Conducting thorough testing to ensure the reliability and performance of the API.
5. Documentation: Creating comprehensive documentation for API users and developers.
6. Deployment: Deploying the API to a production environment for public or restricted access.

Key Features

The Recipes API offers the following key features:

- **Data Retrieval:** Users can retrieve recipe data for specific category, ingredients, and sources.
- **Data Submission:** Authenticated users can submit new recipe data or update existing records.
- **Authentication:** Secure authentication mechanisms, such as API keys or OAuth, ensure controlled access to the API.
- **Integration Support:** The API supports easy integration with external applications, enabling developers to incorporate emission data seamlessly.
- **Documentation:** Comprehensive API documentation provides guidelines for usage and integration.

Use Cases

The CO_2 Emission Utility API can be employed in various use cases, including:

- Meal Planning Assistant: users can use the API to to provide them with a diverse range of recipes suitable for specific meals, dietary requirements, or nutritional goals.
- Smart Kitchen Gadgets Integration: Users can select a recipe from the app, and the connected device will automatically set the cooking parameters, temperatures, and timers .
- Culinary Education Platforms: Developers can integrate the API into educational tools to Enhance course materials with practical examples, and allow students to access a wide range of recipes, cooking tips, and video demonstrations directly from the platform.
- Nutrition and Dietary Apps: The API offer users access to recipes tailored to their dietary preferences, calorie goals, or health conditions.

Conclusion

The Recipes API project aims to be a valuable resource for accessing and managing recipe data. By providing a user-friendly interface, robust authentication, and extensive documentation, the API supports a wide range of use cases in Meal Planning Assistant, Smart Kitchen Gadgets Integration, education, and Nutrition and Dietary initiatives.

3 Token Generation Endpoint

The token generation endpoint (‘/token‘) is a crucial component of the project's utility, residing within the "Authentication" category. This endpoint is designed to facilitate secure access to protected resources by generating access tokens based on user credentials. The primary objective is to establish a secure and authenticated communication channel between the client and the API.

3.1 Endpoint Summary

Endpoint Path: `/token`
Summary: Generate Access Token
Tags: Authentication

3.2 OAuth 2.0 Grant Type

The `/token` endpoint utilizes the OAuth 2.0 Resource Owner Password Credentials (ROPC) grant type for authentication. In this grant type, the client sends a `POST` request with the user's credentials provided in the request body. The required parameters, `username` and `password`, are submitted using the `form_data` parameter.

3.3 Request Parameters

The `/token` endpoint expects a `POST` request with the following parameters in the request body, utilizing the `form_data` parameter:

- `username`: The username of the user.
- `password`: The password associated with the given username.

Upon successful authentication, the endpoint responds with a JSON object containing the generated access token and the token type. The response structure is as follows:

- `access_token`: The generated access token.
- `token_type`: The type of the token, typically "bearer" in accordance with OAuth 2.0 standards.

3.4 Example Response

An example response from the '/token' endpoint:

```
{
  "access_token": "user123token",
  "token_type": "bearer"
}
```

This response signifies that the user with the username "user123" has been successfully authenticated, and an access token has been generated for subsequent secure API requests.

3.5 Security Considerations

- **HTTPS Usage:** It is imperative to ensure that the '/token' endpoint operates over HTTPS to encrypt communication and protect user credentials during transmission.
- **Credential Protection:** The API must securely store and handle user credentials to prevent unauthorized access. Proper password hashing and encryption techniques should be employed.
- **Token Expiry:** Access tokens should have a limited validity period, and mechanisms for token refresh or reauthorization should be implemented to enhance security.
- **Scope Restrictions:** The API should enforce proper scope restrictions based on the user's role and permissions to control access to specific resources.

3.6 Conclusion

The '/token' endpoint serves as a critical element in ensuring secure authentication within the CO_2 Emission Utility API. By adhering to OAuth 2.0 standards and incorporating robust security measures, this endpoint establishes a foundation for controlled and authenticated access to protected resources, contributing to the overall security posture of the API.

4 RecipeData Schema

The ‘RecipeData‘ schema serves as a fundamental structure for representing recipe preparation within the API. This schema is designed to capture key parameters associated with each recipe, providing a standardized format for data input and output.

4.1 Schema Definition

The ‘RecipeData‘ schema is defined as follows:

```
class RecipeIn(BaseModel):
    title: str A string representing the title of the recipe.
    ingredients: str A string representing the list of ingredients.
    servings: str A string representing the number of servings.
    instructions: str A string representing the cooking instructions.

class RecipeOut(BaseModel):
    id: int A unique identifier of each recipe
    recipe: RecipeIn
```

This schema inherits from the ‘BaseModel‘ class provided by the FastAPI framework, ensuring compatibility with the API’s data validation and serialization features. It comprises the following fields:

- `id`: The identifier of the recipe , represented as an int.
- `title`: The title of the recipe, represented as a string.
- `ingredients`: The ingredient to make the recipe, represented as a string.
- `servings`: The number of people to be served, represented as a string.
- `instructions`: The instructions to be followed , represented as a string.

Each field is accompanied by a description providing clarity on its intended purpose and data type.

4.2 Usage in API Endpoints

The ‘RecipeData‘ schema plays a central role in the API’s create, update, and retrieval operations. It is utilized in the following contexts:

- **Create Recipe Endpoint:** When a new atmospheric CO_2 estimation is submitted through the API, the request body is expected to conform to the structure defined by the ‘Estimation-Data‘ schema.
- **Update Recipe Endpoint:** When updating an existing estimation, the ‘EstimationData‘ schema is used to validate and structure the updated data.
- **Response Model:** The ‘EstimationData‘ schema is employed as part of the response model for API endpoints, ensuring consistent and well-structured output when retrieving atmospheric CO_2 estimations.

4.3 Conclusion

The schema serves as a crucial component in maintaining consistency and integrity in the representation of food recipe within the Utility API. By providing a clear and standardized structure, it enhances data validation, facilitates documentation, and promotes a coherent approach to handling estimation data across various API functionalities.

5 GET RECIPE Endpoint

The 'GET RECIPE' endpoint is a crucial component of the CO2 Emission Utility API, residing within the "Recipes" category. This endpoint is designed to retrieve recipes from an external data source and provide a well-structured response to clients.

5.1 Endpoint Summary

Endpoint Path: `/get_recipe`
Summary: Get recipes from an external API.
Tags: Recipes

5.2 Functionality

The primary functionality of the 'GET ESTIMATIONS' endpoint is to fetch atmospheric CO_2 estimations from an external API. The endpoint requires an access token, obtained through authentication, to ensure secure access to the estimations data.

Upon receiving a request, the endpoint makes a 'GET' request to the external API, which provides daily atmospheric CO_2 concentration values. The obtained data is then structured into a list of dictionaries, with each dictionary containing CO_2 values for a specific day.

5.3 Request Parameters

- `token`: A required parameter representing the access token obtained through authentication. This ensures that only authorized users can access the estimations data.

5.4 Response Model

The response model for the 'GET RECIPE' endpoint is defined as follows:

```
List[Dict[str, float]]
```

This signifies that the endpoint returns a list of dictionaries, where each dictionary contains CO_2 values for a specific day. The values are represented as floating-point numbers.

5.5 External API Integration

The endpoint integrates with an external API hosted at the following URL:

```
https://api-ninjas.com/api/recipe
```

The API requires specific headers, including the X-RapidAPI-Key and X-RapidAPI-Host, to authenticate and authorize the requests.

5.6 Conclusion

The ‘GET RECIPE‘ endpoint provides a streamlined mechanism for retrieving recipes from an external data source. By integrating with this endpoint, clients can access up-to-date and accurate recipes, contributing to the overall functionality and value proposition of the RECIPES Utility API.

6 GET RECIPES Endpoint

The endpoint is a crucial element of the Utility API, residing within the ”Recipes” category. This endpoint is designed to retrieve the recipes stored in the global variable and provide a well-structured response to clients.

6.1 Endpoint Summary

Endpoint Path: `/recipes`
Summary: Get recipes
Tags: recipes

6.2 Functionality

The primary functionality of the ‘GET RECIPES‘ endpoint is to return a list of recipes stored in the global variable. To access this endpoint, clients must provide a valid access token obtained through authentication.

6.3 Request Parameters

- `token`: A required parameter representing the access token obtained through authentication. This ensures that only authorized users can access the estimations data.

6.4 Response Model

The response model for the ‘GET RECIPES‘ endpoint is defined as follows:

```
List[Dict[str, Any]]
```

This signifies that the endpoint returns a list of dictionaries, where each dictionary contains recipes for a specific category. The values within each dictionary are represented as generic data types.

6.5 Response Description

The response description for the ‘GET RECIPES‘ endpoint is:

```
"List of Recipes"
```

This provides context for the returned data, indicating that the response contains a recipes.

6.6 Error Handling

The endpoint includes error handling to catch any exceptions that may occur during execution. In case of an error, the API responds with an HTTP 500 Internal Server Error, providing details in the response body.

6.7 Conclusion

The ‘GET RECIPES‘ endpoint facilitates the retrieval of recipes stored in the global variable. By adhering to authentication requirements and providing a well-structured response, this endpoint supports the API's objective of delivering accurate and accessible recipe data to authorized clients.

7 CREATE RECIPE Endpoint

The ‘CREATE RECIPE‘ endpoint is a vital component of Recipes Utility API, residing within the "Recipes" category. This endpoint enables the creation of recipe and returns an updated list of recipes.

7.1 Endpoint Summary

Endpoint Path: `/recipe`
Summary: Create New recipe
Tags: recipes

7.2 Functionality

The primary functionality of the ‘CREATE RECIPE‘ endpoint is to allow users to submit data for a new recipe. To access this endpoint, clients must provide a valid access token obtained through authentication.

The endpoint expects a request body containing the data for the new estimation. This data is validated against the ‘RecipeData‘ schema, and upon successful validation, the recipe is appended to the global list of recipes.

7.3 Request Parameters

- `token`: A required parameter representing the access token obtained through authentication. This ensures that only authorized users can create new recipes.
- `estimation_data`: The request body containing the data for the new recipe. This data is validated against the ‘RecipeData‘ schema.

7.4 Response Model

The response model for the ‘CREATE RECIPE‘ endpoint is defined as follows:

```
List[Dict[str, Any]]
```

This signifies that the endpoint returns a list of dictionaries, where each dictionary contains recipes for a specific category. The values within each dictionary are represented as generic data types.

7.5 Response Description

The response description for the ‘CREATE RECIPES‘ endpoint is:

```
"List of updated RECIPES"
```

7.6 Error Handling

The endpoint includes error handling to catch any exceptions that may occur during execution. In case of an error, the API responds with an HTTP 500 Internal Server Error, providing details in the response body.

7.7 Conclusion

The ‘CREATE RECIPE‘ endpoint facilitates the submission of recipe. By adhering to authentication requirements, validating incoming data, and providing a well-structured response, this endpoint supports the API's objective of enabling users to contribute to the repository of estimations in a secure and controlled manner.

8 Update Recipe Endpoint

The ‘Update RECIPE‘ endpoint is a critical component of the CO2 Emission Utility API, located within the "Recipes" category. This endpoint facilitates the modification of recipe and returns the updated information.

8.1 Endpoint Summary

Endpoint Path: `/recipe`
Summary: Update recipe
Tags: Recipes

8.2 Functionality

The primary functionality of the ‘Update recipe‘ endpoint is to allow users to modify the data of an existing recipe. Access to this endpoint requires a valid access token obtained through authentication.

The endpoint expects a request body containing the updated data for the recipe. This data is validated against the ‘RecipeData‘ schema. If the specified estimation exists in the global list, the data is updated, and the updated estimation is returned.

8.3 Request Parameters

- `token`: A required parameter representing the access token obtained through authentication. This ensures that only authorized users can update recipe.
- `estimation_data`: The request body containing the updated data for recipe. This data is validated against the ‘RecipeData‘ schema.

8.4 Response Model

The response model for the ‘Update RECIPE‘ endpoint is defined as follows:

```
Dict[str, Any]
```

This signifies that the endpoint returns a dictionary containing recipe data for a specific category. The values within the dictionary are represented as generic data types.

8.5 Response Description

The response description for the ‘Update RECIPE‘ endpoint is:

```
"Updated Recipe"
```

8.6 Error Handling

The endpoint includes error handling to catch any exceptions that may occur during execution. If the specified recipe is not found, the API responds with an HTTP 404 Not Found error. In case of any other error, an HTTP 500 Internal Server Error is returned, providing details in the response body.

8.7 Conclusion

The ‘Update Recipes‘ endpoint supports the modification of existing recipes. By adhering to authentication requirements, validating incoming data, and providing a well-structured response, this endpoint enhances the API's capabilities in managing and updating recipe data securely.

9 DELETE recipe Endpoint

The ‘DELETE ESTIMATION‘ endpoint is a crucial feature of the Recipes Utility API, situated within the "Recipes" category. This endpoint allows users to remove an existing recipe and returns information about the deleted estimation.

9.1 Endpoint Summary

Endpoint Path: `/recipe`
Summary: Delete recipe
Tags: Recipes

9.2 Functionality

The primary functionality of the ‘DELETE RECIPE‘ endpoint is to enable users to delete an existing recipe. Access to this endpoint requires a valid access token obtained through authentication.

The endpoint expects a request body containing data id identifying the recipe to be deleted. If a matching estimation is found in the global list, it is removed, and the updated list is returned.

9.3 Request Parameters

- `token`: A required parameter representing the access token obtained through authentication. This ensures that only authorized users can delete estimations.
- `recipe_data`: The request body containing data (title,ingredients,servings,instructions) of the recipe to be deleted.

9.4 Response Model

The response model for the ‘DELETE RECIPE‘ endpoint is defined as follows:

```
Dict[str, Any]
```

9.5 Response Description

The response description for the ‘DELETE RECIPE‘ endpoint is:

```
"Deleted Recipe"
```

9.6 Error Handling

The endpoint includes error handling to catch potential exceptions during execution. If the specified recipe is not found, the API responds with an HTTP 404 Not Found error. If there is any other error, an HTTP 500 Internal Server Error is returned, providing details in the response body.

9.7 Debugging Information

For debugging purposes, the endpoint prints the state of the global estimations list before and after the deletion process, aiding in identifying any potential issues during development.

9.8 Conclusion

The ‘DELETE RECIPE‘ endpoint supports the removal of existing recipe. By adhering to authentication requirements, validating incoming data, and providing a well-structured response, this endpoint enhances the API's capabilities in managing and deleting recipe data securely.

9.9 Requirements.txt

The requirements.txt file is a critical component of the CO2 Emission Utility API project, specifying the external Python packages and their versions required for the proper functioning of the application.

9.9.1 Requirements Content

```
fastapi==0.68.0
uvicorn==0.15.0
httpx==0.21.0
requests==2.26.0
```

9.9.2 Explanation

- **fastapi==0.68.0:** FastAPI is a modern, fast web framework for building APIs with Python. The specified version 0.68.0 ensures compatibility with the API project.
- **uvicorn==0.15.0:** Uvicorn is an ASGI server that serves FastAPI applications. The version 0.15.0 is specified to match the API project's requirements.
- **httpx==0.21.0:** HTTPX is a fully featured HTTP client for Python. Version 0.21.0 is specified to ensure compatibility with the API project.
- **requests==2.26.0:** Requests is a popular HTTP library for Python. The specified version 2.26.0 is included to meet the project's HTTP-related requirements.

9.9.3 Conclusion

The requirements.txt file clearly lists the dependencies and their versions needed for the CO2 Emission Utility API project. This file is crucial for ensuring consistency in package versions across different environments and facilitating smooth deployment and execution of the FastAPI application.

References

1. Python Software Foundation. (2022). *The Python programming language.* Retrieved from https://www.python.org/
2. FastAPI Documentation. (2022). *FastAPI Documentation.* Retrieved from https://fastapi.tiangolo.com/
3. Uvicorn Documentation. (2022). *Uvicorn Documentation.* Retrieved from https://www.uvicorn.org/
4. HTTPX Documentation. (2022). *HTTPX Documentation.* Retrieved from https://www.python-httpx.org/