Universität St.Gallen

University of St.Gallen – School of Management, Economics, Law, Social Sciences, International Affairs and Computer Science (HSG)

Smart Cookbook Project

Python – Introduction to Programming

Gabriele Bortolin (22-613-822)

Maxime Gubelmann (20-610-424)

Yasmin Nishino (20-616-926)

Kevin Strepparava (22-611-727)

Universität St.Gallen

Project

Skills: Programming – Introduction Level

Prof. Dr. Mario Silic

December 21st, 2023

1 What the project does

This document describes a Python code serving as potential Recipe Finder Application. The application is designed to interact with the Spoonacular API to fetch recipes based on user-provided ingredients and to output detailed recipes including nutrition data, if required. Overall, it demonstrates the functions of an Application Programming interface, and the basic interactions with it, such as the authentication of HTTP requests, and the definition of parameters to retrieve desired information.

The Spoonacular API provides various information about "food ontology", including the individual recipes ingredients, significant nutritional data, preparation process and dietary needs or relevant intolerances (Urbansky, n.d.-a).

1.1 Why is this project useful

Most people, especially students that are new to the game of living by their own, may struggle to plan groceries in advance and end up surrendering to impulsive purchases that only result in a filled refrigerator, often with mismatched random ingredients that are hard to combine into one appealing recipe. Meanwhile, those who do put effort in grocery planning may lack inspiration after some time or would like to minimize time efforts. Also, individuals with special dietary needs and allergies face significant obstacles in following a healthy customized diet, they might need to seek professional nutritionist advice which can be costly.

This program, aims to tackle all these problems by promoting healthy customized practices which minimize trips to the grocery shop, decrease food waste, foster cooking creativity, and enable free educational insights on nutritional contents. By offering an extensive selection of various recipes fitting every need, it facilitates meal planning and saves valuable time and money.

2 Checklist for users to get started with the project

To make use of the code, the following prerequisites must be checked out:

- Access to a Python distribution package with interactive Jupyter Python Notebook
- Open file:
    - Cookbook_project.ipynb –Jupyter notebooks
- Valid API secret key provided separately:
    - Spoonacular enables free access to API keys which have limited request capacity.

3 Running the code

This project interacts with an API accessed via a Web resource through Hypertext Transfer Protocol, the latter enables to query data through GET method and it returns a response from the specified URL based on the provided parameters.

3.1 HTTP Library

To send GET HTTP requests the "request" library must be imported which facilitates the usage of multiple methods, like get, post, delete, requests and head from the specified URL (*Python Requests Module*, n.d.).

```
import requests #to make HTTP requests from a specified URL
from requests.exceptions import HTTPError, ConnectionError #to handle unsuccessful responses
```

When interacting with an API through HTTP GET requests the delivered responses might not always be successful, therefore the **.exceptions** module contains several classes to handle these unsuccessful responses.

- HTTPError is raised when the request reaches the server, but the server is not able to smoothly process it and returns an error which could be due to several issues. Examples, include Client Error responses or Server Error responses.
- ConnectionError is raised when there are problems related to connecting with the server.
- Timeout is raised whenever the response is not processed within a certain time frame (GeeksforGeeks, 2023).

3.2 API & API key

The Spoonacular API provides an encrypted key which enables user authentication, to prevent misuse and control the number of interactions with the API (*What Is an API Key? | API Key Definition | Fortinet*, n.d.). Coder's best practice is not to disclose the secret key within the hardcode, therefore the key is provided separately for security reasonws, and it must be copied and pasted into the following code line:

```
# paste the API key provided in a separate file or a separate source
api_key = 'insert the key'
```

Once the key is provided the secret token is passed by the application to enable the user identification.

4 HTTP GET request & Response

4.1 Get request & API documentation

```
def get_recipe_information(recipe_id, api_key, include_nutrition=True):
```

The function **get_recipe_information(),** is at the heart of the program. It enables interaction with the API, which normally starts from a client sending a GET request to the server, with certain specifications. This server-client communication only works with pre-defined rules and protocols and usually most APIs provide a specific unique documentation which illustrates possible requests and parameters

that can be queried (Hall, 2023). The Spoonacular API's documentation provides numerous query possibilities and customizable specifications, such as nutritional information about the recipes. It further provides clear and detailed instructions on the functions, endpoints, and responses, enabling a smooth and easier usage of the API (Urbansky, n.d.-b).

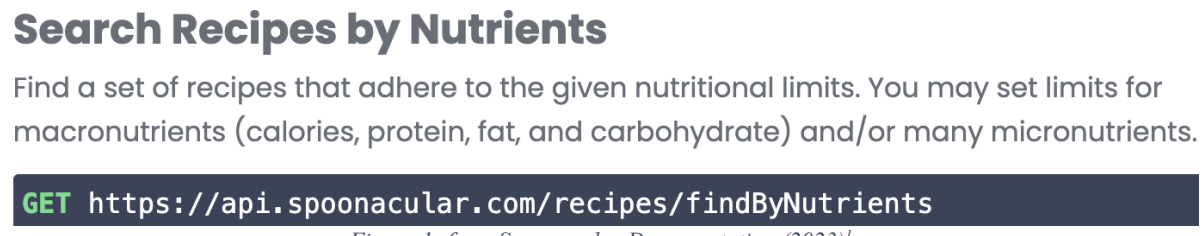

Figure 1: from Spoonacular Documentation (2023)[1]

Thus, within our program, the recipe_id parameter uniquely identifies the recipe, the api key will grant access to Spoonacular's data and include_nutrition is set to true meaning the request to search for recipes will always require including nutritional data as well unless otherwise specified.

4.2 Response & Error handling

This code block within the **get_recipe_information()** function, executes and processes a response from the server. Responses are the result of a correct communication process, which needs to be designed to be understood by the server API and require a correctly formulated specification. For this purpose, the URL, known as the endpoint, specified in the API's documentation, is stored in **info_url** variable which inserts the queried parameters for the search into the placeholders, and functions as address in which to look for the parameters. The response will be stored in the variable **response** which calls the get method of the request's library to send a get request to the specified endpoint and parameters stored in **info_url**.

```
try:
    include_nutrition_str = 'true' if include_nutrition else 'false'
    info_url = f"https://api.spoonacular.com/recipes/{recipe_id}/information?includeNutrition={include_nutrit

    response = requests.get(info_url)
    response.raise_for_status()

    return response.json()

except HTTPError as http_err:
    print(f"HTTP error occurred: {http_err}")
except ConnectionError as conn_err:
    print(f"Connection error occurred: {conn_err}")
except Timeout as timeout_err:
    print(f"Timeout error occurred: {timeout_err}")
except Exception as err:
    print(f"An error occurred: {err}")

return None
```

Note 1: image retrieved from: https://spoonacular.com/food-api/docs#Search-Recipes-by-Nutrients

If the response is successfully processed by the server it will conventionally result in JavaScript Object Notation, i.e .JSON response. Its compatibility with web services, and human readability, enables JSON data to be interpreted and generated in any programming language (*What Is JSON*, n.d.).

As mentioned before, the responses for the queries might not always be successful as several errors may occur in the process. The **try** and **except** are used to handle these possible errors which may occur during a client-server communication such as network connection errors, protocol communication errors, impossibility to retrieve the queried data and other errors which will return **None** to the function's call.

4.3 Get request: find by ingredients

The second formulated get request is fundamentally structured in the same manner, but it differs in its query, to the extent that it communicates with a different endpoint, with different parameters, to retrieve a maximum number of recipes based on ingredients.

```
ingredients = input("Enter the ingredients you have available (separated by commas): ")
find_url = f"https://api.spoonacular.com/recipes/findByIngredients?ingredients={ingredients}&number=5&apiKey={api_key}"
```

The new endpoint's placeholders now include ingredients prompted from the user in the **ingredients'** variable, and the number of recipes to be displayed.

5 Input from user

```
# Ask the user if they want to include nutrition data
include_nutrition = input("Would you like to include nutrition data? (yes/no): ").lower() == 'yes'

# Fetch recipes based on user input
recipes = get_recipes_by_ingredients(api_key)

# Check if recipes were found
if recipes:
    print("\nHere are some recipes based on the ingredients you have:")

    for recipe in recipes:
        recipe_info = get_recipe_information(recipe['id'], api_key, include_nutrition)

        if recipe_info:
            print(f"\nRecipe: {recipe_info['title']}")
            print(f"Link: {recipe_info.get('sourceUrl', 'No URL available')}")
            print(f"Missing Ingredients: {', '.join([ingredient['name'] for ingredient in recipe['missedIngredien

            if include_nutrition and 'nutrition' in recipe_info:
                print("Nutritional Information (per serving):")
                for nutrient in recipe_info['nutrition']['nutrients']:
                    print(f"- {nutrient['name']}: {nutrient['amount']}{nutrient['unit']}")
            else:
                print("Nutritional information is not included.")
else:
    print("No recipes found. Please try different ingredients.")
```

5.1 Asking user input

```
# Ask the user if they want to include nutrition data
include_nutrition = input("Would you like to include nutrition data? (yes/no): ").lower() == 'yes'
```

This part of the code is meant to give flexibility to the user, letting him decide if the nutrition data should be inserted in the recipe search. The input function takes the string text as a parameter and displays a message asking for a positive or negative response suggested in the question (yes or no). To avoid the input being case-sensitive, the **lower()** method converts all the strings into lower case letters. The result is then saved as a binary variable that can have one of two possibilities values in the variable **include_nutrition** (**True** if the user input is equal to the string "yes", **False** otherwise).

5.2 Fetching recipes

```
# Fetch recipes based on user input
recipes = get_recipes_by_ingredients(api_key)
```

This section recalls the API to recover recipes using the ingredients mentioned by the user. The method **get_recipes_by_ingredients()** is then executed and it stores also the results in **recipes** variable. This function uses the ingredients that are supplied by the user and uses them as filters to obtain a recipe that results from a combination of the input and the missing ingredients.

5.3 Displaying Recipe Information

```
# Check if recipes were found
if recipes:
    print("\nHere are some recipes based on the ingredients you have:")

    for recipe in recipes:
        recipe_info = get_recipe_information(recipe['id'], api_key, include_nutrition)

        if recipe_info:
            print(f"\nRecipe: {recipe_info['title']}")
            print(f"Link: {recipe_info.get('sourceUrl', 'No URL available')}")
            print(f"Missing Ingredients: {', '.join([ingredient['name'] for ingredient in recipe['missedIngredien

            if include_nutrition and 'nutrition' in recipe_info:
                print("Nutritional Information (per serving):")
                for nutrient in recipe_info['nutrition']['nutrients']:
                    print(f"- {nutrient['name']}: {nutrient['amount']}{nutrient['unit']}")
            else:
                print("Nutritional information is not included.")
else:
    print("No recipes found. Please try different ingredients.")
```

In this section the code checks if there are any recipes with the ingredients entered by the user. Through **"if receipes:"**, it starts a loop to compare each recipe with the ingredients entered. To obtain further information, it calls **get_recipe_information()**, and if the recipe has information (which is **if recipe_info**), the code publishes the relevant title, the URL and in case they are missing, also the missing ingredients. If the user has selected to add nutrition information (so if **include_nutrition** is **True**), the code checks if nutritional data is available into **recipe_info** and, if positive, the code provides the nutritional information for the recipe. Otherwise, if this information is not available, the code generates a message which informs the user that the requested information was not found.
This section of code is in charge of retrieving, showing and interacting with recipe information based on user's requests (input), considering the availability of recipes and additional information.

6 Disclosure of aids & sources

GeeksforGeeks. (2023, January 23). *Exception Handling of Python Requests module*. https://www.geeksforgeeks.org/exception-handling-of-python-requests-module/

Hall, D. (2023, December 12). *How to use an API: Just the basics*. TechnologyAdvice. https://technologyadvice.com/blog/information-technology/how-to-use-an-api/

Python Requests module. (n.d.). https://www.w3schools.com/python/module_requests.asp

Urbansky, D. (n.d.). *spoonacular recipe and food API*. https://spoonacular.com/food-api/docs#Search-Recipes-by-Nutrients

Urbansky, D. (n.d.-b). *spoonacular recipe and food API*. https://spoonacular.com/food-api/docs#Analyze-Recipe

What is JSON. (n.d.). https://www.w3schools.com/whatis/whatis_json.asp https://www.w3schools.com/whatis/whatis_json.asp

What is an API key? | API key Definition | Fortinet. (n.d.). Fortinet. https://www.fortinet.com/resources/cyberglossary/api-key#:~:text=An%20application%20programming%20interface%20(API,secret%20token%20for%20authentication%20purposes

The project was developed with assistance of OpenAI's ChatGPT to provide guidance on python, debugging and suggestions. All final coding, report, commenting, analysis, and underlying code idea was performed by us in accordance with ethical guidelines and academic integrity.

Declaration of authorship

"I hereby declare

- that I have written this thesis without any help from others and without the use of documents and aids other than those stated above;
- that I have mentioned all the sources used and that I have cited them correctly according to established academic citation rules;
- that I have acquired any immaterial rights to materials I may have used such as images or graphs, or that I have produced such materials myself;
- that the topic or parts of it are not already the object of any work or examination of another course unless this has been explicitly agreed on with the faculty member in advance and is referred to in the thesis;
- that I will not pass on copies of this work to third parties or publish them without the University's written consent if a direct connection can be established with the University of St.Gallen or its faculty members;
- that I am aware that my work can be electronically checked for plagiarism and that I hereby grant the University of St.Gallen copyright in accordance with the Examination Regulations in so far as this is required for administrative action;
- that I am aware that the University will prosecute any infringement of this declaration of authorship and, in particular, the employment of a ghostwriter, and that any such infringement may result in disciplinary and criminal consequences which may result in my expulsion from the University or my being stripped of my degree."

St.Gallen, 20. 12 2023

Yasmin Nishino  Kevin Strepparava 

……

By submitting this academic term paper, I confirm through my conclusive action that I am submitting the Declaration of Authorship, that I have read and understood it, and that it is true.