Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра ІСМ

Звіт
про виконання лабораторної роботи № 7
«Робота з API та веб-сервісами»
з дисципліни
«Спеціалізовані мови програмування»

Виконав:
Студент групи ІТ-32,
Вольвенко І. Р.

Прийняв:
Щербак С.С

Львів 2023

Мета роботи: Створення консольного об’єктно - орієнтованого додатка з використанням API

Завдання:

Завдання 1: Вибір провайдера API

Виберіть надійний API, який надає через HTTP необхідні дані для віддаленого зберігання, вивантаження або реалізуйте свій. Для прикладу це може бути jsonplaceholder.org

Завдання 2: Інтеграція API

Виберіть бібліотеку для роботи з API та обробки HTTP запитів (для прикладу це може бути бібліотека Requests). Інтегруйте обраний API в ваш консольний додаток на Python. Ознайомтеся з документацією API та налаштуйте необхідний API-ключ чи облікові дані.

Завдання 3: Введення користувача

Розробіть користувальницький інтерфейс, який дозволяє користувачам візуалізувати всі доступні дані в табличному вигляді та у вигляді списку. Реалізуйте механізм для збору та перевірки введеного даних користувачем.

Завдання 4: Розбір введення користувача

Створіть розбірник для видобування та інтерпретації виразів користувача на основі регулярних виразів, наприклад, для візуалізації дат, телефонів, тощо. Переконайтеся, що розбірник обробляє різні формати введення та надає зворотний зв'язок про помилки.

Завдання 5: Відображення результатів

Реалізуйте логіку для візуалізації даних через API в консолі. Обробляйте відповіді API для отримання даних у вигляді таблиць, списків. Заголовки

таблиць, списків мають виділяться кольором та шрифтом, які задається користувачем

Завдання 6: Збереження даних

Реалізуйте можливості збереження даних у чіткому та читабельному форматі JSON, CSV та TXT

Завдання 7: Обробка помилок

Розробіть надійний механізм обробки помилок для керування помилками API, некоректним введенням користувача та іншими можливими проблемами. Надавайте інформативні повідомлення про помилки.

Завдання 8: Ведення історії обчислень

Включіть функцію, яка реєструє запити користувача, включаючи введені запити та відповідні результати. Дозвольте користувачам переглядати та рецензувати історію своїх запитів.

Завдання 9: Юніт-тести

Напишіть юніт-тести для перевірки функціональності вашого додатку. Тестуйте різні операції, граничні випадки та сценарії помилок.

Код:

user_service.py:

```
import json
import requests
from prettytable import PrettyTable
import regex
import config
from config import logger

class UserService:
```

```
    @staticmethod
    def get_personal_profile(twitter_username: str):
        if not twitter_username or not isinstance(twitter_username, str):
            raise ValueError("Username must be a non-empty string!")

        query_params = {"username": twitter_username}
        headers = {
            "X-RapidAPI-Key": config.X_RAPID_API_KEY,
            "X-RapidAPI-Host": config.X_RAPID_API_HOST
        }
        response = requests.get(config.GET_PERSONAL_PROFILE, headers=headers,
params=query_params)
        if response.status_code != 200:
            message = response.json().get('message', 'Unknown error occurred!')
            logger.error("Error: %s", message)
            raise ValueError(f"Error occurred! {message}")

        return response.json()

class DisplayInTableService:

    @staticmethod
    def display_personal_profile(json_data: str):
        data = json.loads(json_data)

        table = PrettyTable()
        table.field_names = ["Attribute", "Value"]

        allowed_keys = {
            "user_id", "username", "name", "follower_count", "following_count",
            "favourites_count", "is_private", "is_verified", "location",
            "profile_pic_url", "profile_banner_url", "description", "external_url",
            "number_of_tweets", "bot"
        }
```

```
        for key, value in data.items():
            if key in allowed_keys:
                table.add_row([key, value])

        return table.get_string()
```

user_menu.py:

```
import json

from config import TWITTER_ACC_INFO_JSON, TWITTER_ACC_INFO_CSV, TWITTER_ACC_INFO_TXT
from service.lab7.user_service import DisplayInTableService, UserService
from shared import json_processor
from shared.color_processor import ColorProcessor
from shared.file_processors import FileProcessor



class UserMenu:

    @staticmethod
    def run():
        history: list = []
        successful_result: bool = False
        jsons: list = []

        while True:
            print("Choose an option:")
            print("1. Display data of a Twitter personal profile")
            print("2. Save data")
            print("3. Show history")
            print("0. Exit")

            option = input("Your choice: ")
            match option:
                case "1":
```

```
    jsons = []
    twitter_username = input("Enter Twitter username: ")
    try:
        jsons = UserService.get_personal_profile(twitter_username)
        print("Choose an option:")
        print("1. Display data in a flattened way")
        print("2. Display data in JSON format")
        print("3. Display data in a table")
        while True:
            option = input("Your choice: ")
            match option:
                case "1":
                    ColorProcessor.display_colors()
                    color_position = int(input("Enter a color position: "))
                    json_processor.display_flattened_json(jsons, color_position)
                    break
                case "2":
                    print(json.dumps(jsons, indent=4))
                    break
                case "3":
                    print(DisplayInTableService.display_personal_profile(
                        json.dumps(jsons, indent=4)))
                    break
                case _:
                    print("Invalid option. Enter again!")
        history.append(
            f"Data of a Twitter personal profile for username "
            f"{twitter_username}:\n{json.dumps(jsons, indent=4)}")
        successful_result = True
    except ValueError as e:
        print(e)
        successful_result = False

case "2":
    if successful_result:
```

```
                try:
                    FileProcessor.write_into_json(TWITTER_ACC_INFO_JSON, jsons)
                    FileProcessor.write_into_csv(TWITTER_ACC_INFO_CSV, jsons)
                    FileProcessor.write_into_txt(TWITTER_ACC_INFO_TXT, jsons)
                    print("Data saved in JSON, CSV, and TXT formats.")
                except Exception as e:
                    print(f"Error saving data: {e}")
            else:
                print("No data to save!")

        case "3":
            if len(history) == 0:
                print("No history!")
            else:
                for counter, item in enumerate(history):
                    print(f"{counter + 1}: {item}")

        case "0":
            break

        case _:
            print("Invalid option. Enter again!")
```

Результат виконання у форматі json:


```
{
    "creation_date": "Tue Jun 02 20:12:29 +0000 2009",
    "user_id": "44196397",
    "username": "elonmusk",
    "name": "Elon Musk",
    "follower_count": 166232481,
    "following_count": 506,
    "favourites_count": 37957,
    "is_private": null,
    "is_verified": false,
    "is_blue_verified": true,
```

```
    "location": "\ud835\udd4f\u00d0",
    "profile_pic_url":
"https://pbs.twimg.com/profile_images/1683325380441128960/yRsRRjGO_normal.jpg",
    "profile_banner_url": "https://pbs.twimg.com/profile_banners/44196397/1690621312",
    "description": "",
    "external_url": null,
    "number_of_tweets": 34934,
    "bot": false,
    "timestamp": 1243973549,
    "has_nft_avatar": false,
    "category": null,
    "default_profile": false,
    "default_profile_image": false,
    "listed_count": 149605,
    "verified_type": null
}
```


Результат виконання у форматі csv:

creation_date,user_id,username,name,follower_count,following_count,favourites_count,is_private,is_verified,is_blue_verified,location,profile_pic_url,profile_banner_url,description,external_url,number_of_tweets,bot,timestamp,has_nft_avatar,category,default_profile,default_profile_image,listed_count,verified_type

Tue Jun 02 20:12:29 +0000 2009,44196397,elonmusk,Elon Musk,166232481,506,37957,,False,True,ХÐ,https://pbs.twimg.com/profile_images/1683325380441128960/yRsRRjGO_normal.jpg,https://pbs.twimg.com/profile_banners/44196397/1690621312,,,34934,False,1243973549,False,,False,False,149605,

Результат виконання у форматі txt:

"creation_date"

"user_id"

"username"

"name"

"follower_count"

"following_count"

"favourites_count"

"is_private"

"is_verified"

"is_blue_verified"

"location"

"profile_pic_url"

"profile_banner_url"

"description"

"external_url"

"number_of_tweets"

"bot"

"timestamp"

"has_nft_avatar"

"category"

"default_profile"

"default_profile_image"

"listed_count"

"verified_type"

Демонстрація виконання програми наведена на рисунку 1:

```
2. Save data
3. Show history
0. Exit
Your choice: 1
Enter Twitter username: elonmusk
Choose an option:
1. Display data in a flattened way
2. Display data in JSON format
3. Display data in a table
Your choice: 3
+--------------------+--------------------------------------------------------------------------------+
|     Attribute      |                                     Value                                      |
+--------------------+--------------------------------------------------------------------------------+
|      user_id       |                                    44196397                                    |
|      username      |                                    elonmusk                                    |
|        name        |                                   Elon Musk                                    |
|   follower_count   |                                   166203605                                    |
|  following_count   |                                      506                                       |
|  favourites_count  |                                     37957                                      |
|     is_private     |                                      None                                      |
|    is_verified     |                                     False                                      |
|      location      |                                       ℵÐ                                       |
|  profile_pic_url   | https://pbs.twimg.com/profile_images/1683325380441128960/yRsRRjGO_normal.jpg |
| profile_banner_url |          https://pbs.twimg.com/profile_banners/44196397/1690621312           |
|    description     |                                                                                |
|    external_url    |                                      None                                      |
|  number_of_tweets  |                                     34934                                      |
|        bot         |                                     False                                      |
+--------------------+--------------------------------------------------------------------------------+
```

Рис. 1 Виконання програми

Посилання на GitHub репозиторій: https://github.com/Deadmarvald/smp

Висновки: Виконавши ці завдання, я створив проект, який надав мені цінний досвід роботи з API, дизайну користувацького інтерфейсу, валідації введення, обробки помилок та тестування.