Міністерство освіти і науки України Харківський національний університет радіоелектроніки

Кафедра ІРТЗІ

Лабораторна робота №1

З навчальної дисципліни

"Мережне програмування"

Виконав:

студент 4 курсу

Групи КСТЗІ 21-1

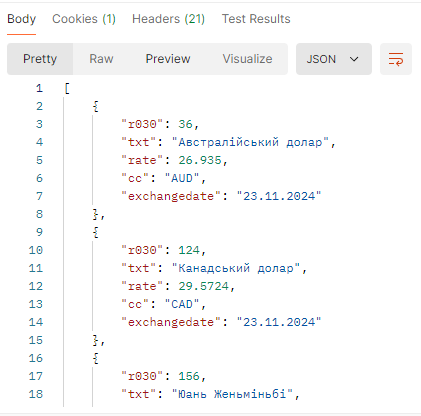
Передерій І.А

Перевірив:

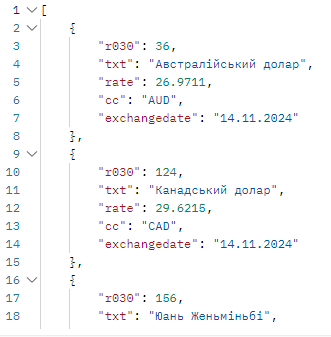
Андрушко Д.В

Харків 2024

1**.    [Easy] Отримати курс валют із сайту НБУ за попередній тиждень за допомогою Postman. Приклад використання API -** [**https://bank.gov.ua/admin\_uploads/article/Instr\_API\_KURS\_VAL\_data.pdf**](https://bank.gov.ua/admin_uploads/article/Instr_API_KURS_VAL_data.pdf)



**Рисунок 1** – Get запит на сайт НБУ за 23.11.2024



**Рисунок 2** – Get запит на сайт НБУ за 23.11.2024

2**. [Easy] Отримати курс валют із сайту НБУ за попередній тиждень з використанням python-бібліотеки requests**

**Код на Pythone**

import requests

from datetime import datetime, timedelta

# Базовий URL для отримання курсу валют

BASE\_URL = "https://bank.gov.ua/NBUStatService/v1/statdirectory/exchange"

# Функція для отримання курсу валют за конкретну дату

def get\_exchange\_rate(date):

    formatted\_date = date.strftime('%Y%m%d')  # Формат YYYYMMDD

    response = requests.get(f"{BASE\_URL}?date={formatted\_date}&json")

    if response.status\_code == 200:

        return response.json()

    else:

        print(f"Error fetching data for {formatted\_date}: {response.status\_code}")

        return None

# Отримання курсу валют за попередній тиждень

def get\_last\_week\_rates():

    today = datetime.now()

    last\_week = [today - timedelta(days=i) for i in range(7)]

    rates = {}

    for date in last\_week:

        rates[date.strftime('%Y-%m-%d')] = get\_exchange\_rate(date)

    return rates

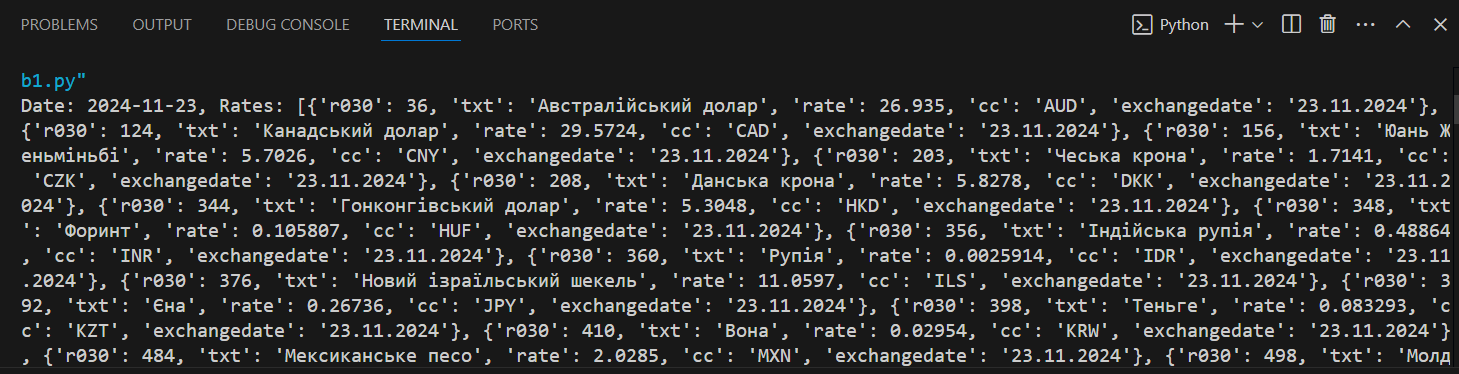
# Виклик функції та друк результатів

rates = get\_last\_week\_rates()

for date, data in rates.items():

    print(f"Date: {date}, Rates: {data}")

**Результат кода**



**Рисунок 3** – курс валют

1. **[Easy-Medium] Побудувати графік зміни курсів валют за допомогою бібліотеки matplotlib**

**Код на Pythone**

# Импорт необходимых библиотек

import requests  # Для отправки HTTP-запросов к API НБУ

from datetime import datetime, timedelta  # Для работы с датами

import matplotlib.pyplot as plt  # Для построения графиков

# Базовый URL API НБУ для получения курсов валют

BASE\_URL = "https://bank.gov.ua/NBUStatService/v1/statdirectory/exchange"

# Функция для получения курсов валют за конкретную дату

def get\_exchange\_rate(date):

    """

    Получить курсы валют за указанную дату.

    :param date: объект datetime, представляющий дату

    :return: JSON-ответ с курсами валют за указанную дату

    """

    formatted\_date = date.strftime('%Y%m%d')  # Преобразуем дату в формат YYYYMMDD

    response = requests.get(f"{BASE\_URL}?date={formatted\_date}&json")  # Отправляем GET-запрос к API

    if response.status\_code == 200:  # Если запрос успешен

        return response.json()  # Возвращаем данные в формате JSON

    else:

        print(f"Ошибка получения данных для {formatted\_date}: {response.status\_code}")

        return None  # Возвращаем None в случае ошибки

# Функция для получения курсов валют за последние 7 дней

def get\_last\_week\_rates():

    """

    Получить курсы валют за последние 7 дней.

    :return: словарь с датами и курсами валют

    """

    today = datetime.now()  # Текущая дата

    last\_week = [today - timedelta(days=i) for i in range(7)]  # Список дат за последние 7 дней

    rates = {}  # Словарь для хранения курсов валют

    for date in last\_week:  # Проходим по каждой дате

        rates[date.strftime('%Y-%m-%d')] = get\_exchange\_rate(date)  # Сохраняем курсы валют по дате

    return rates  # Возвращаем словарь с курсами валют

# Функция для построения графика изменения курса валют

def plot\_currency\_change(rates, currency\_code="USD"):

    """

    Построить график изменения курса валюты.

    :param rates: словарь с датами и курсами валют

    :param currency\_code: код валюты, для которой строится график (по умолчанию USD)

    """

    dates = []  # Список для хранения дат

    currency\_rates = []  # Список для хранения значений курса валюты

    for date, data in rates.items():  # Проходим по всем датам и данным из словаря

        dates.append(date)  # Добавляем дату в список

        # Находим курс для указанного кода валюты

        rate = next((item['rate'] for item in data if item['cc'] == currency\_code), None)

        if rate:  # Если курс найден

            currency\_rates.append(rate)  # Добавляем курс в список

    # Построение графика

    plt.figure(figsize=(10, 5))  # Размер графика

    plt.plot(dates, currency\_rates, marker='o', label=currency\_code, color='blue')  # Линия графика с маркерами

    plt.title(f"Зміна курсу {currency\_code} за останній тиждень", fontsize=14)  # Заголовок графика

    plt.xlabel("Дата", fontsize=12)  # Подпись оси X

    plt.ylabel("Курс (грн)", fontsize=12)  # Подпись оси Y

    plt.grid(True)  # Включаем сетку

    plt.legend(fontsize=12)  # Добавляем легенду

    plt.xticks(rotation=45)  # Поворачиваем метки оси X для удобства

    plt.tight\_layout()  # Автоматическая подгонка графика

    plt.show()  # Показываем график

# Основной блок кода

if \_\_name\_\_ == "\_\_main\_\_":

    # Получаем данные о курсах валют за последние 7 дней

    rates = get\_last\_week\_rates()

    # Проверяем, удалось ли загрузить данные

    if rates:

        # Строим график для валюты USD (доллар США)

        plot\_currency\_change(rates, currency\_code="USD")

    else:

        print("Не удалось загрузить данные о курсах валют.")

**Результат**

**** **Рисунок 4** – график доллара

****

**Рисунок 5** – график євро