Модель проєкта на основі JSON-даних веб-серверів

Постановка задачі

Національний банк України надає щодня дані про курси валют. Дані про курси можуть бути в форматі JSON (а також в форматі XML). Побудувати проєкт пакету функцій аналізу банківських даних.

Проєкт може складатись з двох частин: 1) отримати від банку нові дані; 2) опрацювати дані за допомогою власних алгоритмів і функцій. Для демонстрації будемо вважати, що функції будуємо на основі одного або двох файлів даних курсів валют.

Загальна будова проєкту може бути такою:

Визначити у власному проєкті місце (папку) зберігання файлів JSON-даних
Перевірити, чи є в папці найновіші дані за останній день
Якщо даних немає – зробити перший запит до сервера
Якшо сервер не відповів – спробувати зробити повторний запит
Повідомити про результат запиту
Виконати дослідження структури json-файла (перевірити структуру)
Прийняти рішення про вибір двох файлів даних
Визначити файл для друкування результатів аналізу даних
Опрацювати дані за результатами одного файла найновіших даних (статичний аналіз)
Опрацювати дані за результатами двох файлів (аналіз динаміки)

Модуль отримання даних від банку

```
#  --------- отримати дані від банку про курси валют ------
#  ----------- автоматизація запитів до сервера -----------

#  використовуємо модуль urllib.request
import urllib.request
import os, sys
import glob
import time, datetime

print("Start . . .")

# файл JSON
remoteaddr =
"https://bank.gov.ua/NBUStatService/v1/statdirectory/exchange/?mode=json"
# файл XML
#remoteaddr =
"https://bank.gov.ua/NBUStatService/v1/statdirectory/exchange"

# ---------------------------------------------------------------
# Функція: перевірити, чи є  файл, отриманий сьогодні (за датою файла)
def TestAvailable(directory):  # перевірка в заданій папці directory
  # час сьогоднішнього дня
  today = str(datetime.date.today().day)+'.' +   \
          str(datetime.date.today().month)   \
          + '.' + str(datetime.date.today().year)
  print("сьогодні:", today)
  maindirname = os.path.abspath('.')  # шлях від кореня до поточної папки
  # об'єднати шлях з папкою
  dirname = os.path.join(maindirname, directory)
  #print(dirname)
  # список файлів типу *.json
```

```
  part = glob.glob(os.path.join(dirname,'*.json'))
  # сортувати файли за датами
  part.sort(key=os.path.getmtime, reverse=True)
  print("наявні такі файли json:")
  print( *[os.path.split(filename)[1]  for filename in part], sep='\n' )
  # [ далі можна спростити, взявши до уваги найперший файл в part -
  #   найновіший ]

  #""" - - - - - - -  варіант 1 (повний) - - - - - - - - - -
  # побудувати список часу створення файлів
  gt = [os.path.getmtime(filename) for filename in part]
  #print(gt)
  # перетворити в структуровану форму:
  sttm = [ time.localtime(file) for file in gt ]
  # локальний час - структура time.struct_time
  #print(sttm)
  # відокремити день, місяць, рік
  dmy = [ str(tmf.tm_mday)+'.'+str(tmf.tm_mon)+'.'   \
          +str(tmf.tm_year) for tmf in sttm ]
  print("є файли за такими датами:", dmy)
  # повернути відповідь і дату останнього файла
  if today in dmy: return (True, today, os.path.split(part[0])[1])
  else: return (False, dmy[0], os.path.split(part[0])[1])
  # найперший файл - найновіший (найбільше getmtime)
  # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -"""

  """ - - - - - - - -  варіант 2 (простіший) - - - - - - - - -
  # локальний час найновішого файла
  sttm = time.localtime(os.path.getmtime(part[0]))
  dmy = str(sttm.tm_mday)+'.'+str(sttm.tm_mon)+'.'+str(sttm.tm_year)
  print("дата найновішого файла:", dmy)
  # повернути відповідь і дату останнього файла
  if today == dmy: return (True, today, os.path.split(part[0])[1])
  else: return (False, dmy, os.path.split(part[0])[1])
  # найперший файл - найновіший (найбільше getmtime)
  # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -"""
  pass
# -------- def TestAvailable(directory) ------------------------

# функція: зробити запит на файл, повідомити про результат
def Request(directory):
  filename = "bank-exc-" + str(datetime.date.today().day)   \
             + '-' + str(datetime.date.today().month) + ".json"
  maindirname = os.path.abspath('.')  # шлях від кореня до поточної папки
  destfile = os.path.join(maindirname, directory, filename)
  #print(destfile)  # ім'я для отриманого файла
  try:
    #  копіювати файл з перетворенням кодування ANSI в UTF-8
    remotefile = urllib.request.urlopen(remoteaddr)
    # об'єкт файла для читання
    with open(destfile,"w") as fsave :   # зберегти у текстовий файл
      fsave.write(remotefile.read().decode(encoding='utf-8'))  # копія
    remotefile.close()
    reply = (True,destfile)
  except:  # можуть бути помилки з'єднання
    info = sys.exc_info()
    # інформація про помилку (type, value, traceback)
    reply = (False, info[0],info[1])
```

```
  return reply
  pass
# -----------def Request(directory)--------------------------------

# всі файли в папці exchangedata
filesdir = "exchangedata"
# перевірити, чи вже є сьогоднішній файл
print(". . . test available . . .")
whetherIsHere = TestAvailable(filesdir)
print("file exist:", whetherIsHere)

#sys.exit()  # припинити виконання сценарію – для налагодження

# тут може бути функція прийняття рішення про отримання файла;
#  треба врахувати, що в окремі дні сервер нових даних не надає,
#  наприклад, на вихідні дні

# зробити запит на файл; у випадку помилки сервера - повторити один раз
if not whetherIsHere[0]:
  print(". . . request . . .")
  answer = Request(filesdir)
  print(answer)
  if not answer[0]:
    print(". . . pause . . .")
    time.sleep(3)  # затримати виконання на 3 секунди
    print(". . . request repeat . . .")
    answer2 = Request(filesdir); print(answer2)


print(". . . finish")

pass  # кінець сценарію
```

Приклади виконання модуля отримання даних

```
Start . . .
. . . test available . . .
сьогодні: 2.12.2019
наявні такі файли json:
bank-exc-1-12.json
bank-exc-25-11.json
є файли за такими датами: ['1.12.2019', '25.11.2019']
file exist: (False, '1.12.2019', 'bank-exc-1-12.json')
. . . request . . .
(False, <class 'urllib.error.URLError'>,
URLError(ConnectionResetError(10054, 'Віддалений хост примусово закрив
наявне підключення', None, 10054, None)))
. . . pause . . .
. . . request repeat . . .
(True, 'D:\\V_V\\Python Lecture\\PythonTest\\L25a\\exchangedata\\bank-
exc-2-12.json')
. . . finish
```

```
Start . . .
. . . test available . . .
сьогодні: 2.12.2019
наявні такі файли json:
bank-exc-2-12.json
bank-exc-1-12.json
bank-exc-25-11.json
є файли за такими датами: ['2.12.2019', '1.12.2019', '25.11.2019']
file exist: (True, '2.12.2019', 'bank-exc-2-12.json')
. . . finish
```

Дослідження структури json-файла (перевірити структуру)

```
# дослідження структури json-файлів на прикладі банківських документів
#  курси обміну валют
import json
import os.path

# всі файли в папці exchangedata
filesdir = "exchangedata"
file1 = 'bank-exc-25-11.json'
file2 = 'bank-exc-2-12.json'
jsonfile1 = os.path.join(filesdir, file1)
jsonfile2 = os.path.join(filesdir, file2)

with open(jsonfile1) as data_file:
  data0212 = json.load(data_file)

with open(jsonfile2) as data_file:
  data2511 = json.load(data_file)

from pprint import pprint
#pprint(data0212)


#""" ------------------------------------------------------
# функція: надрукувати параметри json-документа
def PrintJsonParam(data):
  print("Тип цілого документа: ", type(data).__name__)
  print("Документ має елементів: ", len(data))
  print("Тип кожного елемента як цілого:\n",   \
        *[type(ob).__name__ for ob in data])
  print("Вигляд одного окремого елемента:\n", data[0])
  print("Ключі окремих частин елемента (словника):\n", *data[0].keys())
  print("Типи ключів і значень частин елемента:",
         *[( type(list(data[0].keys())[k]).__name__, " : ",
          type(list(data[0].values())[k]).__name__ )
          for k in range(len(data[0])) ], sep='\n')
  print(40*"-")
  pass
# ------------------------------------------------------ """

print("файл",file1)
PrintJsonParam(data0212)
print("файл",file2)
PrintJsonParam(data2511)

# будемо вважати, що структура всіх елементів json-документа є однакова;
#   в загальному випадку потрібно перевірити структуру кожного елемента
```

Приклад результатів дослідження (фрагмент)

```
файл bank-exc-2-12.json
Тип цілого документа:  list
Документ має елементів:  61
Тип кожного елемента як цілого:
 dict dict dict dict dict dict dict dict dict dict dict dict dict dict
dict dict dict dict dict dict dict dict dict dict dict dict dict dict
dict dict dict dict dict dict dict dict dict dict dict dict dict dict
dict dict dict dict dict dict dict dict dict dict dict dict dict dict
dict dict dict dict dict
Вигляд одного окремого елемента:
 {'r030': 36, 'txt': 'Австралійський долар', 'rate': 16.222796,
  'cc': 'AUD', 'exchangedate': '02.12.2019'}
Ключі окремих частин елемента (словника):
 r030 txt rate cc exchangedate
Типи ключів і значень частин елемента:
('str', ' : ', 'int')
('str', ' : ', 'str')
('str', ' : ', 'float')
('str', ' : ', 'str')
('str', ' : ', 'str')
----------------------------------------
```

Модуль статичного аналізу (за даними одного файла)

```
# модуль статичного аналізу даних (за результатами одного файла)
# ------- обрати файл JSON-даних для аналізу --------
import json
import os.path
import datetime

# всі файли в папці filesdir
filesdir = "exchangedata"
file1 = 'bank-exc-2-12.json'   # json-файл даних
jsonfile1 = os.path.join(filesdir, file1)

report = "report.txt"   # файл результатів аналізу
filereport = os.path.join(filesdir, report)
f = open(filereport,'w')   # або  open(filereport,'a')

with open(jsonfile1) as data_file:
  data0212 = json.load(data_file)

from pprint import pprint
#pprint(data0212)

listdata = []  # список друкування результатів
# час сьогоднішнього дня
today = str(datetime.date.today().day)+'.' +
str(datetime.date.today().month)   \
        + '.' + str(datetime.date.today().year)
listdata.append("Звіт за: " + today)

#     Загальні характеристики json-документа
#  Тип цілого документа:  list
#  Документ має елементів:  61
#  Тип кожного елемента як цілого: dict
#  Вигляд одного окремого елемента
# {'r030': 36, 'txt': 'Австралійський долар', 'rate': 16.222796,
#   'cc': 'AUD', 'exchangedate': '02.12.2019'}
#  Ключі окремих частин елемента (словника):
#       r030 txt rate cc exchangedate
#  Типи ключів і значень частин елемента:
#  ('str', ' : ', 'int')
#  ('str', ' : ', 'str')
#  ('str', ' : ', 'float')
#  ('str', ' : ', 'str')
#  ('str', ' : ', 'str')

#""" -------- Показати 10 найвищих обмінних курсів ---------
# сортувати список за полем 'rate' кожного словника даних
data = data0212[:]  # копія всіх даних
data.sort(key=lambda R: R['rate'], reverse=True)
listdata.append("{0:^46}".format("10 НАЙВИЩИХ ОБМІННИХ КУРСІВ"))
listdata.append(55*'-')
listdata.append("| {0:<34}| {1:<11}|".  \
                format("Назва валюти", "Курс обміну за 1"))
listdata.append(55*'-')
for k in range(10):
  listdata.append("| {0:<34}| {1:<11}грн. |".  \
                  format(data[k]['txt'], data[k]['rate']))
```

```
listdata.append(55*'-')
#"""

#""" -------- Показати 10 найвищих курсів лише для валют ---------
data = data0212[:]  # копія всіх даних
data.sort(key=lambda R: R['rate'], reverse=True)
# викреслити з списку елементи з такими кодами 'r030':
#      964, 959, 962, 961, 960

def removecode(code): # викреслити елемент заданого коду
  i = -1
  for k in range(len(data)):
    if data[k]['r030']==code:  i=k; break
  if i>=0: del data[i:i+1]

removecode(964); removecode(959); removecode(962); removecode(961);
removecode(960)

listdata.append("{0:^46}".format("10 НАЙВИЩИХ КУРСІВ ВАЛЮТ"))
listdata.append(54*'-')
listdata.append("| {0:<34}| {1:<11}|".  \
                format("Назва валюти", "Курс обміну за 1"))
listdata.append(54*'-')
for k in range(10):
  listdata.append("| {0:<34}| {1:<11}грн. |".  \
                  format(data[k]['txt'], data[k]['rate']))
listdata.append(54*'-')
#"""

#""" ------ Які валюти відрізняються від гривні не більше 0.5 ---------
data = data0212[:]  # копія всіх даних
data.sort(key=lambda R: R['rate'], reverse=True)
listdata.append("{0:^45}".format("КУРСИ ЩОДО ГРИВНІ (+-)0.5"))
listdata.append(54*'-')
listdata.append("| {0:<29}| {1:<9}| {2:<9}|".  \
                format("Назва валюти", "Курс", "Різниця"))
listdata.append(54*'-')
for k in range(len(data)):
  if abs(data[k]['rate']-1.0)<= 0.5 :
    listdata.append("| {0:<29}| {1:<9}| {2:<+9}|".  \
          format(data[k]['txt'], round(data[k]['rate'], 2),  \
                 round(data[k]['rate']-1.0, 2)))
listdata.append(54*'-')
#"""

# всі результати на екран
for i in range(len(listdata)) : print(listdata[i])
for line in listdata:  f.write(line + '\n')  # всі результати в файл
f.close()  # закрити файл результатів

pass
```

Приклад результатів статичного аналізу

```
Звіт за: 2.12.2019
         10 НАЙВИЩИХ ОБМІННИХ КУРСІВ
-------------------------------------------------------
| Назва валюти                       | Курс обміну за 1|
-------------------------------------------------------
| Паладій                            | 43941.182  грн. |
| Золото                             | 34912.024  грн. |
| Платина                            | 21335.326  грн. |
| Срібло                             | 406.21     грн. |
| СПЗ(спеціальні права запозичення)  | 32.911545  грн. |
| Фунт стерлінгів                    | 30.890412  грн. |
| Євро                               | 26.326354  грн. |
| Долар США                          | 23.972276  грн. |
| Швейцарський франк                 | 23.937401  грн. |
| Канадський долар                   | 18.014475  грн. |
-------------------------------------------------------
           10 НАЙВИЩИХ КУРСІВ ВАЛЮТ
-------------------------------------------------------
| Назва валюти                       | Курс обміну за 1|
-------------------------------------------------------
| Фунт стерлінгів                    | 30.890412  грн. |
| Євро                               | 26.326354  грн. |
| Долар США                          | 23.972276  грн. |
| Швейцарський франк                 | 23.937401  грн. |
| Канадський долар                   | 18.014475  грн. |
| Сінгапурський долар                | 17.531034  грн. |
| Лівійський динар                   | 17.066977  грн. |
| Австралійський долар               | 16.222796  грн. |
| Новозеландський долар              | 15.404537  грн. |
| Азербайджанський манат             | 14.101339  грн. |
-------------------------------------------------------
          КУРСИ ЩОДО ГРИВНІ (+-)0.5
-------------------------------------------------------
| Назва валюти                  | Курс     | Різниця  |
-------------------------------------------------------
| Єгипетський фунт              | 1.49     | +0.49    |
| Молдовський лей               | 1.37     | +0.37    |
| Мексіканський песо            | 1.23     | +0.23    |
| Чеська крона                  | 1.03     | +0.03    |
| Бат                           | 0.79     | -0.21    |
| Новий тайванський долар       | 0.79     | -0.21    |
-------------------------------------------------------
```

Модуль аналізу динаміки курсів (за даними двох файлів)

```
# модуль аналізу динаміки даних (за результатами двох файлів)
# ------- обрати файли JSON-даних для аналізу --------
import json
import os.path, os
import datetime, time

# всі файли в папці filesdir
filesdir = "exchangedata"
file1 = 'bank-exc-2-12.json'   # json-файл даних
jsonfile1 = os.path.join(filesdir, file1)
file2 = 'bank-exc-25-11.json'   # json-файл даних
jsonfile2 = os.path.join(filesdir, file2)


report = "report2.txt"   # файл результатів аналізу
filereport = os.path.join(filesdir, report)
f = open(filereport,'w')   # або  open(filereport,'a')

with open(jsonfile1) as data_file:
  data0212 = json.load(data_file)
with open(jsonfile2) as data_file:
  data2511 = json.load(data_file)

from pprint import pprint
#pprint(data0212)

# час створення файлів
sttm1 = time.localtime(os.path.getmtime(jsonfile1))
dmy1 = str(sttm1.tm_mday)+'.'+str(sttm1.tm_mon)+'.'+str(sttm1.tm_year)
sttm2 = time.localtime(os.path.getmtime(jsonfile2))
dmy2 = str(sttm2.tm_mday)+'.'+str(sttm2.tm_mon)+'.'+str(sttm2.tm_year)

#     Загальні характеристики json-документа
#  Тип цілого документа:  list
#  Документ має елементів:  61
#  Тип кожного елемента як цілого: dict
#  Вигляд одного окремого елемента
# {'r030': 36, 'txt': 'Австралійський долар', 'rate': 16.222796,
#  'cc': 'AUD', 'exchangedate': '02.12.2019'}
#  Ключі окремих частин елемента (словника):
#       r030 txt rate cc exchangedate
#  Типи ключів і значень частин елемента:
#  ('str', ' : ', 'int')
#  ('str', ' : ', 'str')
#  ('str', ' : ', 'float')
#  ('str', ' : ', 'str')
#  ('str', ' : ', 'str')

listdata = []  # список друкування результатів
# час сьогоднішнього дня
today = str(datetime.date.today().day)+'.' +
str(datetime.date.today().month)   \
        + '.' + str(datetime.date.today().year)
listdata.append("Звіт за:  " + today)
```

```
#""" -------- Динаміка зміни окремих валют валют ---------
# 'r030': Долар США (840), Євро (978), Канадський долар (124)
#         Швейцарський франк (756), Фунт стерлінгів (826), Злотий (985)
def findrateexch(dt,code):  # знайти в dt дані про валюту code
  i = -1
  for k in range(len(dt)):
    if dt[k]['r030']==code:  i=k; break
  if i>=0: return dt[i]
  else: return None

listdata.append("\nДинаміка зміни окремих валют")
listdata.append(" {0:<28}| {1:<12}| {2:<12}| {3:<10}".  \
                format("Назва валюти", dmy2, dmy1, "Різниця"))
listdata.append(67*'-')

for currency in [840,978,124,756,826,985]:
  fr1 = findrateexch(data2511,currency)
  fr2 = findrateexch(data0212,currency)
  if fr1 and fr2:
    listdata.append(" {0:<28}| {1:<12}| {2:<12}| {3:<+10}".  \
         format(fr1['txt'], round(fr1['rate'],2),  \
             round(fr2['rate'],2), round(fr2['rate']-fr1['rate'], 2)))

listdata.append(67*'-')
#"""

#""" ---------- Валюти, які найбільше змінили курс --------------
listdata.append("\nВалюти, які найбільше змінили курс за різницею")
listdata.append(" {0:<28}| {1:<12}| {2:<12}| {3:<10}| {4:<10}".  \
           format("Назва валюти", dmy2, dmy1, "Різниця", "Відсоток"))
listdata.append(79*'-')
parttable = list(zip( [code['r030'] for code in data0212],
                      [cap['txt'] for cap in data0212],
                      [dprev['rate'] for dprev in data2511],
                      [dnext['rate'] for dnext in data0212]))
parttable.sort(key=lambda x: abs(x[2]-x[3]), reverse=True)
# викреслити з списку банківські метали і СПЗ
#    (коди 'r030': 964, 959, 962, 961, 960)
def removecode(code): # викреслити елемент заданого коду
  i = -1
  for k in range(len(parttable)):
    if parttable[k][0]==code:  i=k; break
  if i>=0: del parttable[i:i+1]

for c in [964, 959, 962, 961, 960]: removecode(c)
#print(*parttable[:10], sep='\n')  # тестова перевірка списку

def percent(x1,x2):  # відсоток |x1-x2| / |x1|
  return round(abs(x1-x2)/abs(x1) * 100, 2)

for k in range(10):  # для прикладу лише 10 валют (для всіх:
len(parttable))
  listdata.append(" {0:<28}| {1:<12}| {2:<12}| {3:<+10}| {4:<10}".
                  format(parttable[k][1],
                         round(parttable[k][2],2),
                         round(parttable[k][3],2),
                         round(parttable[k][3]-parttable[k][2],2),
                         percent(parttable[k][3],parttable[k][2])))
```

```
listdata.append(79*'-')
#"""

for i in range(len(listdata)) : print(listdata[i])  # всі результати на
екран
for line in listdata:  f.write(line + '\n')  # всі результати в файл
f.close()  # закрити файл результатів

pass
```

Приклад результатів аналізу динаміки

Звіт за: 4.12.2019

Динаміка зміни окремих валют

Назва валюти	25.11.2019	2.12.2019	Різниця
Долар США	24.16	23.97	-0.19
Євро	26.72	26.33	-0.39
Канадський долар	18.2	18.01	-0.19
Швейцарський франк	24.3	23.94	-0.37
Фунт стерлінгів	31.08	30.89	-0.19
Злотий	6.22	6.1	-0.12

Валюти, які найбільше змінили курс за різницею

Назва валюти	25.11.2019	2.12.2019	Різниця	%
Лівійський динар	17.72	17.07	-0.65	3.84
Білоруський рубль	11.79	11.37	-0.42	3.68
Туніський динар	8.77	8.37	-0.41	4.85
Євро	26.72	26.33	-0.39	1.5
Швейцарський франк	24.3	23.94	-0.37	1.54
Ларі	8.39	8.07	-0.31	3.88
Туркменський новий манат	7.09	6.85	-0.24	3.53
Ганських седі	4.58	4.34	-0.24	5.54
Дирхам ОАЕ	6.76	6.53	-0.23	3.54
Малайзійський ринггіт	5.94	5.74	-0.2	3.52

.