



LineNotify推播

LINE & Notify

簡介

現代的人很常使用通訊軟體來傳遞公事或私事，有些資訊希望能定時接收，例如每日早上一進公司就可以收到昨日營收資訊、早上出門前可以先收到今日氣象、股市收盤資訊...等等，此時就可以使用**LINE+Notify**功能，將你所想要知道的資訊定時傳送至**LINE**中，隨時掌握最新狀態。



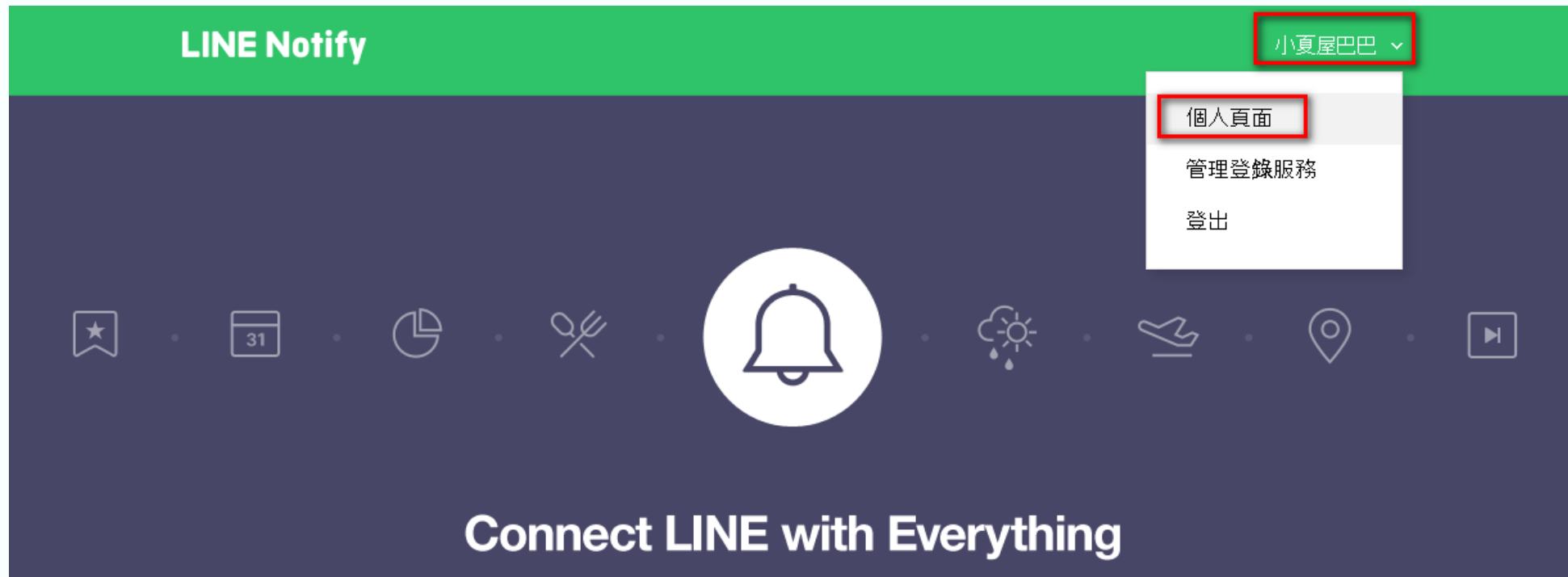
LINE Notify與Message的差異



功能項目	LINE Notify	LINE Message
費用	免費	免費&付費
訊息	只能推播	1.推播 2.快速回覆 3.使用者回覆
聊天室	共用LINE Notify官方聊天室	有自己的聊天室
大頭照	不能更換	可以自訂
主要應用	訊息推播	利用chatbot和使用者互動

創建LINE Notify step.1

- ◆ 至[LINE Notify](#)官網登入LINE帳號
- ◆ 接著再點選「你的帳號」→「我的頁面」



創建LINE Notify step.2

- ◆ 選擇「發行存取權杖(開發人員用)」
- ◆ 填寫權杖名稱及要接收通知的聊天室

發行存取權杖(開發人員用)

若使用個人存取權杖, 不須登錄網站服務, 即可設定通知。

發行權杖

LINE Notify API Document

發行權杖

請填寫權杖名稱(將於傳送提醒時顯示)

匯率通知

請選擇您要接收通知的聊天室。

Search by group name



透過1對1聊天接收LINE Notify的通知

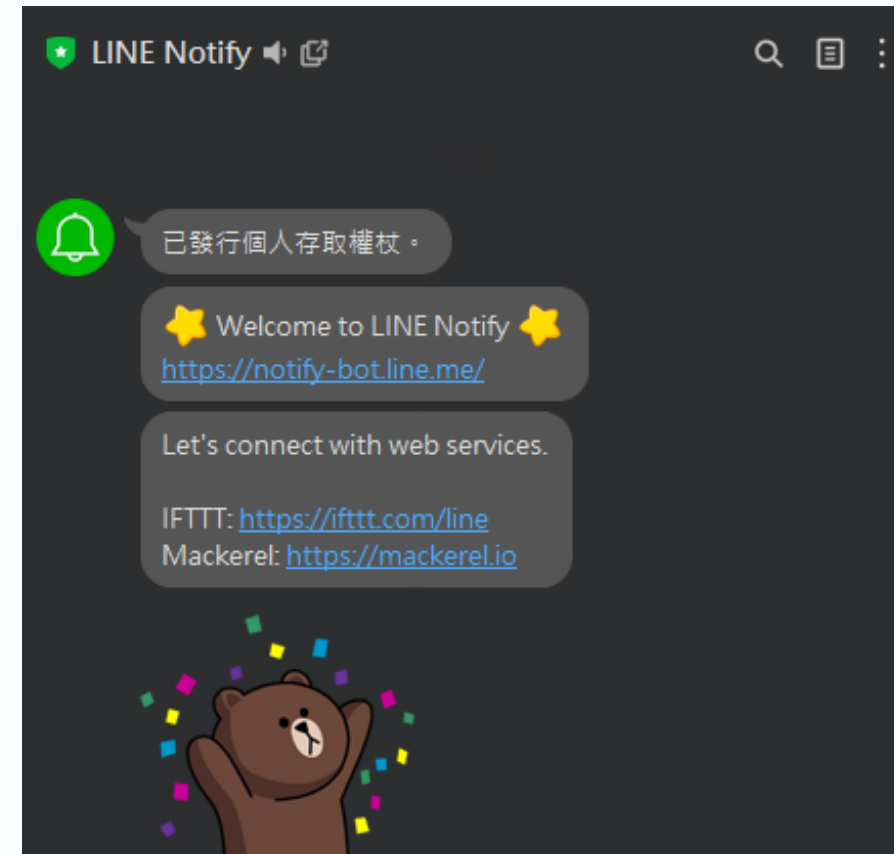


※若公開個人存取權杖, 第三者將能取得您所連動的聊天室名稱及個人資料上的姓名。

發行

創建LINE Notify step.3

- ◆ 完成上一個步驟後系統會提供一個權杖(token)給你，請先複製好畫面上出現的token，並且會在LINE裡收到通知訊息






創建LINE Notify step.4

- ◆ 回到個人頁面可以看到目前已連動的服務
- ◆ 若忘記token，可以解除目前已連動的服務再重新操作步驟2~3取得token

已連動的服務

此為已連動的服務一覽。若想解除連動，請按「解除」鍵。



2022.10.25 13:23

從 匯率通知
傳送至 小夏屋巴巴

[解除](#)

推播文字訊息到LINE

- ◆ 使用request的POST方法向LINE Notify API伺服器送出請求
- ◆ LINE Notify伺服器會從HTTP表頭中取得token來辨識發送請求的對象
- ◆ 欲發送的訊息要放在參數message裡

```
1 import requests
2
3 # =====
4 #   Notify設定
5 # =====
6
7 def notify(msg, token):
8     url = "https://notify-api.line.me/api/notify"           # Notify網址
9     headers = {"Authorization": "Bearer " + token}          # HTTPS表頭
10    payload = {"message": message}                          # HTTPS內容
11    requests.post(url, headers=headers, params=payload)      # 提出POST請求
12
13
14 # 發送訊息
15 token = "lU8denUmV7tpy3qUBF6l1qIqgPDPmrzItLej8CuvKxo"    # 你的token
16 message = "今天天氣真好！"
17 notify(message, token)
```



推播貼圖到LINE

- ◆ 若想要推播時加上LINE貼圖，則需在payload裡設定貼圖編號

```
1  import requests
2
3  # =====
4  #      Notify設定
5  # =====
6
7  def notify(msg, token):
8      url = "https://notify-api.line.me/api/notify"           # Notify網址
9      headers = {"Authorization": "Bearer " + token}          # HTTPS表頭
10     payload = {"message": message,                          # HTTPS內容
11                "stickerPackageId": 11538,                   # 該組貼圖編號
12                "stickerId": 51626496}                       # 指定貼圖編號
13     requests.post(url, headers=headers, params=payload)      # 提出POST請求
14
15
16 # 發送貼圖
17 token = "lU8denUmV7tpy3qUBF611qIqgPDPmrzItLej8CuvKxo"    # 你的token
18 message = "你好"
19 notify(message, token)
```



推播圖片到LINE

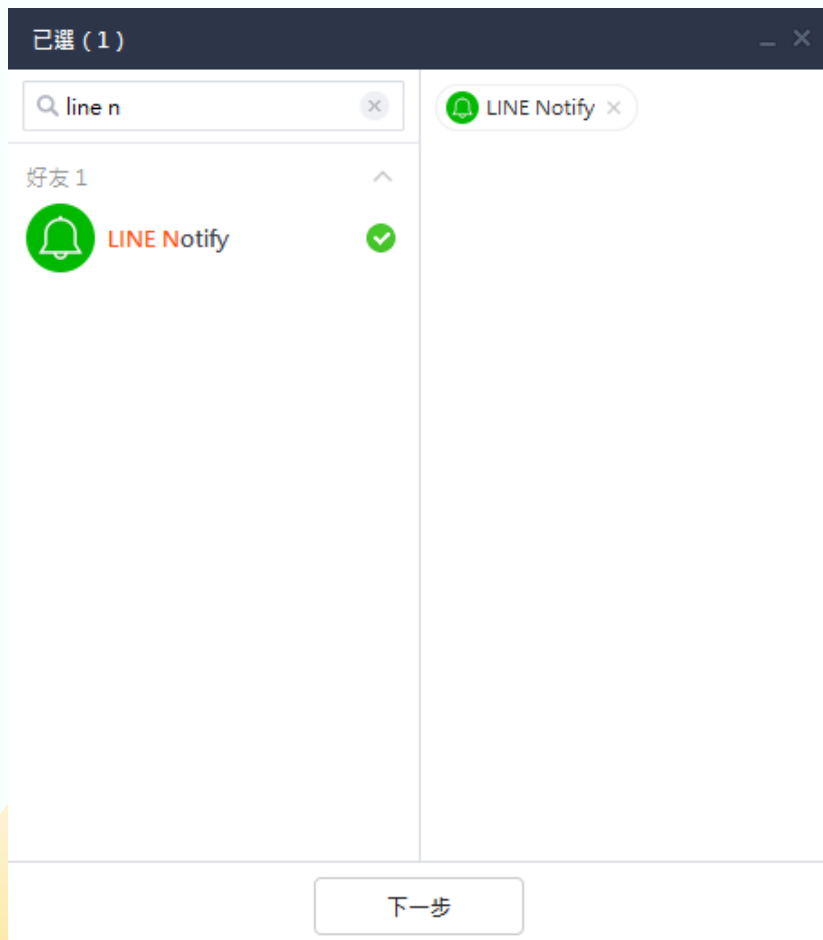
```
1 import requests
2
3 # =====
4 #     Notify設定
5 # =====
6
7 def notify(msg, token, image):
8     url = "https://notify-api.line.me/api/notify"
9     headers = {"Authorization": "Bearer " + token}
10    payload = {"message": message}
11    image = open(image, "rb")
12    imageFile = {"imageFile": image}
13    requests.post(url, headers=headers,
14                  data=payload, files=imageFile)
15
16
17 # 發送圖片
18 token = "lU8denUmV7tpy3qUBF6l1qIqgPDPmrzItLej8CuvKxo"
19 message = "歡迎來到匯率群"
20 img = "cat.jpg"
21 notify(message, token, img)
```

Notify網址
HTTPS表頭
HTTPS內容
設定圖片來源
提出POST請求
你的token



推播訊息到群組 step.1

- ◆ 先創立一個群組(LINE Notify記得要邀請進群組中)





推播訊息到群組 step.2

◆ 選擇發行權杖，並填寫權杖名稱及選擇要接收通知的群組

LINE Notify小夏屋巴巴 ▾

已連動的服務

此為已連動的服務一覽。若想解除連動，請按「解除」鍵。

  2022.10.25 13:31
從 匯率通知
傳送至 小夏屋巴巴 解除

發行存取權杖(開發人員用)

若使用個人存取權杖，不須登錄網站服務，即可設定通知。

發行權杖

[LINE Notify API Document](#)

發行權杖

請填寫權杖名稱(將於傳送提醒時顯示)

台銀匯率通知

請選擇您要接收通知的聊天室。

 工程師測試群

※若公開個人存取權杖，第三者將能取得您所連動的聊天室名稱及個人資料上的姓名。

發行

推播訊息到群組 step.3

- ◆ 將畫面上顯示的權杖(token)複製
- ◆ 確認目前已連動的服務是否有剛剛設定好的群組

已發行的權杖如下。

lu [redacted] Hx

若離開此頁面，將不會再顯示新發行的權杖。離開頁面前，請先複製權杖。

複製

關閉

已連動的服務

此為已連動的服務一覽。若想解除連動，請按「解除」鍵。

	>		2022.10.25 18:31 從 台銀匯率通知 傳送至 工程師測試群	解除
	>		2022.10.25 13:31 從 匯率通知 傳送至 小夏屋巴巴	解除

推播訊息到群組 step.4

- ◆ 將剛拿到的token放入程式碼中，執行程式後群組內的人都可以看到此訊息

```
1  import requests
2
3  # =====
4  #     Notify設定
5  # =====
6
7  def notify(msg, token):
8      url = "https://notify-api.line.me/api/notify"          # Notify網址
9      headers = {"Authorization": "Bearer " + token}         # HTTPS表頭
10     payload = {"message": message,                         # HTTPS內容
11                "stickerPackageId": 8525,                   # 該組貼圖編號
12                "stickerId": 16581296}                     # 指定貼圖編號
13     requests.post(url, headers=headers, params=payload)    # 提出POST請求
14
15
16 # 發送貼圖
17 token = "IuDkSryAcAwlBvhKNNKjQITsXEHjrYwLJeSXHxmC9b0"  # 你的token
18 message = "歡迎來到工程師群"
19 notify(message, token)
```



綜合應用(一)

◆ 用爬蟲至台灣銀行將當日匯率的以下資訊抓取

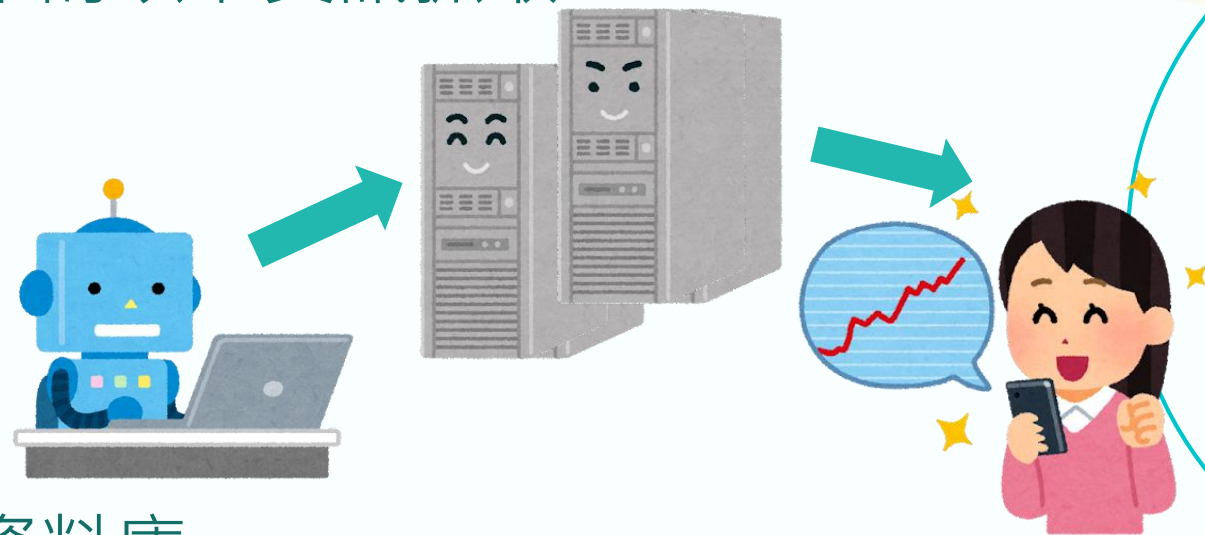
◆ 幣別

◆ 現金匯率 – 本行買入

◆ 現金匯率 – 本行賣出

◆ 將爬蟲抓取資料清理後寫入資料庫

◆ 再從資料庫裡撈出前一日美金的買入及賣出價，用python將資料推播至LINE裡



綜合應用(一) 爬蟲抓取台銀匯率寫入資料庫

```
1 import requests
2 from bs4 import BeautifulSoup
3 import pandas as pd
4 import re
5 from datetime import date
6 import MySQLdb
7
8 # =====
9 # 爬台銀匯率
10 # =====
11 url = "https://rate.bot.com.tw/xrt?Lang=zh-TW"
12 html = requests.get(url)
13 soup = BeautifulSoup(html.text, "lxml")
14
15 table = soup.find("table", "table table-striped table-bordered table-condensed table-hover")
16 # 抓取幣別名稱
17 currency = table.find_all("div", {"class": "hidden-phone print_show"})
18 # 抓取現金買入及賣出價
19 bankbuy = table.find_all("td", {"class": "rate-content-cash text-right print_hide"})
20
21 # =====
22 # 將爬下資料寫成dataframe
23 # =====
24 # 將幣別名稱、買入及賣出分別做成dataframe
25 data1 = pd.DataFrame([(currency[i].text).strip() for i in range(len(currency))], columns=["幣別"])
26 data2 = pd.DataFrame((bankbuy[i].text for i in range(len(bankbuy)) if i%2==0), columns=["本行買入"])
27 data3 = pd.DataFrame((bankbuy[i].text for i in range(len(bankbuy)) if i%2!=0), columns=["本行賣出"])
28
29 # 將三份資料合併(以欄合併)
30 alldata = pd.concat([data1, data2, data3], axis=1)
31 # 將匯率中非數值的值全更換為0
32 alldata["本行買入"] = alldata["本行買入"].replace("-", 0).astype(float)
33 alldata["本行賣出"] = alldata["本行賣出"].replace("-", 0).astype(float)
34
35 # 取得今日日期
36 today = date.today()
37
```

```
38
39 # =====
40 # 將資料寫入資料庫
41 # =====
42
43 try:
44     # 開啟資料庫連接
45     conn = MySQLdb.connect(host="localhost",      # 主機名稱
46                             user="marin",        # 帳號
47                             password="1qazxsw2", # 密碼
48                             database = "testdb1", # 資料庫
49                             port=3306,           # port
50                             charset="utf8")      # 資料庫編碼
51
52     # 使用cursor()方法操作資料庫
53     cursor = conn.cursor()
54
55     # 將資料data寫到資料庫中
56     try:
57         for i in range(len(alldata)):
58             sql = """INSERT INTO taiwanbank_currency (date, currency, buy, sold)
59                     VALUES (%s, %s, %s, %s)"""
60             var = (today, alldata.iloc[i,0], alldata.iloc[i,1], alldata.iloc[i,2])
61             cursor.execute(sql, var)
62
63     conn.commit()
64     print("資料寫入完成")
65
66 except Exception as e:
67     print("錯誤訊息:", e)
68
69 except Exception as e:
70     print("資料庫連接失敗:", e)
71
72 finally:
73     conn.close()
74     print("資料庫連線結束")
75
```


綜合應用(一) 從資料庫中撈取前一日的美金匯率

```
1 import MySQLdb
2 import pandas as pd
3 from datetime import date
4 import datetime
5
6 # =====
7 # 到資料庫找出前一日美金買入及賣出匯率
8 # =====
9
10 def show_usd_currency():
11     # 取得昨日日期
12     today = date.today()
13     yesterday = today + datetime.timedelta(days = -1)
14
15     try:
16         # 開啟資料庫連接
17         conn = MySQLdb.connect(host="localhost", # 主機名
18                                user="marin", # 帳號
19                                password="1qazxsw2", # 密碼
20                                database = "testdb1", # 資料庫
21                                port=3306, # port
22                                charset="utf8") # 資料庫編碼
23
24         # 使用cursor()方法操作資料庫
25         cursor = conn.cursor()
```

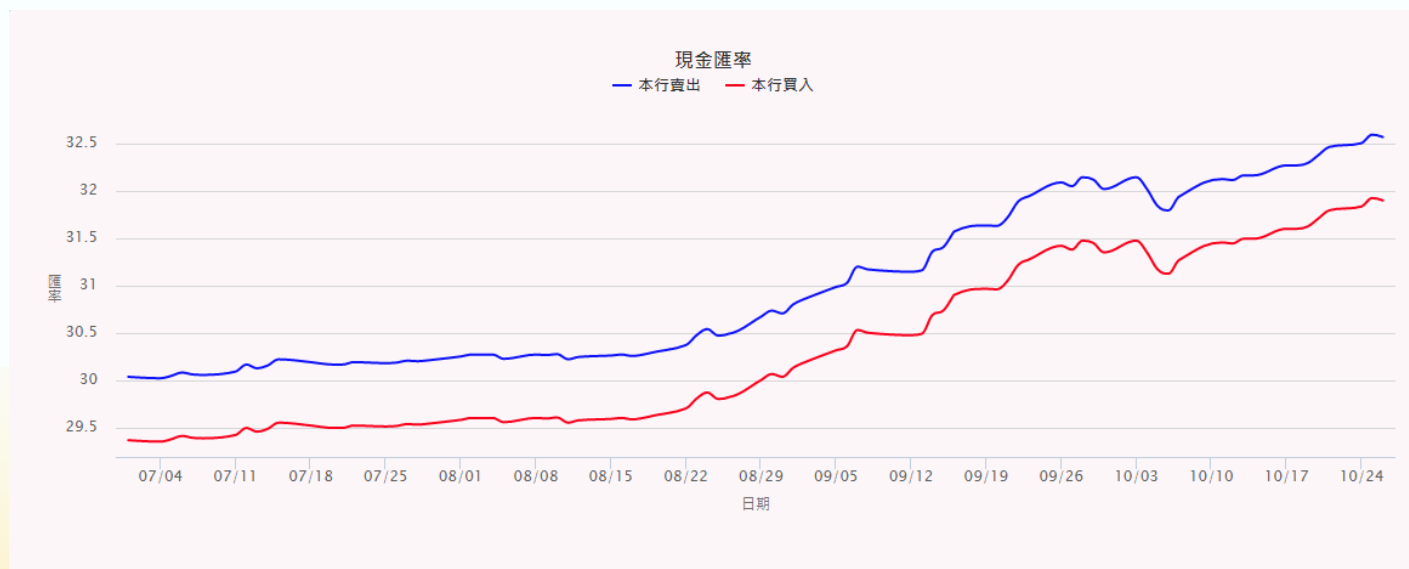
```
27         # 查詢表格taiwanbank_currency的美金買入及賣出價
28     try:
29         sql = """SELECT buy, sold FROM taiwanbank_currency
30                 WHERE currency LIKE '%s' and date='%s'"""
31                 %(("美金"+"%"), yesterday)
32         cursor.execute(sql)
33         data = cursor.fetchone()
34
35     return data
36
37     except Exception as e:
38         print("錯誤訊息:", e)
39
40
41     except Exception as e:
42         print("資料庫連接失敗:", e)
43
44     finally:
45         conn.close()
46         print("資料庫連線結束")
```

綜合應用(一) 將美金匯率推播至LINE

```
1  import requests
2  import webcrawler2
3  from datetime import date
4  import datetime
5
6  # =====
7  #     Notify設定
8  # =====
9
10 def notify(msg, token):
11     url = "https://notify-api.line.me/api/notify"          # Notify網址
12     headers = {"Authorization": "Bearer " + token}         # HTTPS表頭
13     payload = {"message": message}                         # HTTPS內容
14     requests.post(url, headers=headers, params=payload)    # 提出POST請求
15
16
17 # 發送訊息
18 token = "lU8denUmV7tpy3qUBF6l1qIqgPDPmrzItLej8CuvKxo"  # 你的token
19 usd_currency = webcrawler2.show_usd_currency()
20 today = date.today()
21 yesterday = today + datetime.timedelta(days = -1)         # 取得昨日日期
22
23 message = "\n%s的美金\n買入為%s，賣出為%s" %(yesterday, usd_currency[0], usd_currency[1])
24 notify(message, token)
```

綜合應用(二)

- ◆ 讀取從台灣銀行下載美金近三個月匯率檔案(csv)
- ◆ 擷取日期、現金買入與賣出價後畫成折線圖
- ◆ 再用python將圖片推播至LINE



綜合應用(二) 讀取資料並畫成折線圖

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 def draw():
5     data = pd.read_csv("ExchangeRate_USD.csv")
6
7     clean_data = pd.DataFrame({"日期": [data.iloc[i, 0] for i in range(len(data)-1, -1, -1)],
8                                "本行買入": [data.iloc[i, 3] for i in range(len(data)-1, -1, -1)],
9                                "本行賣出": [data.iloc[i, 13] for i in range(len(data)-1, -1, -1)]})
10
11     # 設定畫布
12     plt.figure(figsize=(16, 6))
13
14     # 設定中文
15     plt.rcParams["font.family"] = "Microsoft JhengHei"
16     plt.rcParams["font.size"] = 12
17     plt.rcParams["axes.unicode_minus"] = False
18
19     # 畫折線圖
20     plt.plot(clean_data.index, clean_data["本行買入"], color="#0E12F9", label="本行買入")
21     plt.plot(clean_data.index, clean_data["本行賣出"], color="#FA001B", label="本行賣出")
```

綜合應用(二) 讀取資料並畫成折線圖

```
23     # 調整x軸標籤
24     label = [(str(clean_data.iloc[i,0])[4:]) for i in range(0,len(clean_data),4)]
25     plt.xticks(range(0,len(clean_data),4), labels=label, rotation=45)
26
27     # 設定x軸與y軸標題
28     plt.xlabel("日期")
29     plt.ylabel("匯\n率", rotation=0)
30
31     # 設定圖例及網格
32     plt.legend(title="現金匯率", ncol=2, loc=9)
33     plt.grid(axis="y")
34
35     # 儲存圖表
36     plt.savefig("bankcurrency.jpg")
37
38     plt.show()
```

綜合應用(二) 將圖片推播至LINE

```
1 import requests
2 from datetime import date
3 import datetime
4 import currencydrawimg
5 import webcrawler2
6
7 # =====
8 #     Notify設定
9 # =====
10
11 currencydrawimg.draw()
12 usd_currency = webcrawler2.show_usd_currency()
13
14 # 取得昨日日期
15 today = date.today()
16 yesterday = today + datetime.timedelta(days = -1)
17
18 def notify(msg, token, image):
19     url = "https://notify-api.line.me/api/notify" # Notify網址
20     headers = {"Authorization": "Bearer " + token} # HTTPS表頭
21     payload = {"message": message} # HTTPS內容
22     image = open(image, "rb")
23     imageFile = {"imageFile": image} # 設定圖片來源
24     requests.post(url, headers=headers, # 提出POST請求
25                   data=payload, files=imageFile)
26
27
28 # 發送圖片
29 token = "lU8denUmV7tpy3qUBF6l1qIqgPDPmrzItLej8CuvKxo" # 你的token
30 message = "\n%s的美金\n買入為%s，賣出為%s" %(yesterday, usd_currency[0], usd_currency[1])
31 img = "bankcurrency.jpg"
32 notify(message, token, img)
```

