

# 元大證券 API\_Python 設定說明

## (Windows、MacOS)

聯絡方式：[yuantaapi.brk@yuanta.com](mailto:yuantaapi.brk@yuanta.com)

中華民國 115 年 03 月 13 日

元大證券

# 目錄

|                               |    |
|-------------------------------|----|
| 一、元大 SparkAPI 說明 .....        | 3  |
| 二、環境建置 .....                  | 4  |
| 1. 下載.NET SDK 安裝包 .....       | 4  |
| 2. 安裝 Python 檔(若已安裝可省略) ..... | 7  |
| 3. pip 安裝 pythonnet 套件 .....  | 10 |
| 三、範例程式操作說明 .....              | 11 |
| <b>(一)Windows</b> .....       | 11 |
| 1. 測試環境憑證匯入 .....             | 11 |
| 2. 主程式 YSendOrder.py .....    | 15 |
| <b>(二)MacOS</b> .....         | 19 |
| 主程式 YSendOrder.py .....       | 19 |
| 四、備註 .....                    | 24 |
| 1. 可能會出現的錯誤 .....             | 24 |
| 2. 方法參數型別 .....               | 30 |

# 一、元大 SparkAPI 說明

本文件提供以 python 語言使用 YuantaSparkAPI 之基本範例，在正式環境進行下單及收委託回報的簡單範例程式，並包含其他功能之使用寫法供參考，協助開發者快速上手。

YuantaSparkAPI 使用方式為讀取 C#元件 YuantaSparkAPI.dll 以執行 API 呼叫，由於 Python 無法直接載入 C#元件，故需使用外部套件達成元件引用。本範例使用 Pythonnet 套件，使 Python 程式可呼叫.NET 元件，達成 API 測試與功能開發。

執行環境需求

1. 建議 Python 版本為 3.8 以上
2. 使用 Pythonnet 套件 (pip install pythonnet)

YSendOrder.py 為使用 Pythonnet 之範例程式，於 Python 環境內安裝即可直接執行 py 檔。

各項說明可參考元大證券 API 操作說明、IO\_Doc 資料夾文件，範例程式中也

有相關註解，提醒您 範例僅供參考不代表效能保證。

# 二、環境建置

## 1. 下載.NET SDK 安裝包

YuantaSparkAPI.dll 為.NET8 開發，請進行.NET8.0 的安裝

依據自身電腦作業系統選擇不同的 SDK 安裝檔

[下載 .NET 8.0 \(Linux、macOS 和 Windows\) | .NET](#)

^ 8.0.23

[版本資訊](#) 最新發行日期 2026年1月13日

建置應用程式 - SDK 

SDK 8.0.417

| 作業系統    | 安裝程式  | 二進位檔   |
|---------|---|--|
| Linux   | <a href="#">套件管理員說明</a>   | <a href="#">Arm32</a>   <a href="#">Arm32 Alpine</a>   <a href="#">Arm64</a>   <a href="#">Arm64 Alpine</a>   <a href="#">x64</a>   <a href="#">x64 Alpine</a> |
| macOS   | <a href="#">Arm64</a>   <a href="#">x64</a>   | <a href="#">Arm64</a>   <a href="#">x64</a>  |
| Windows | <a href="#">x64</a>   <a href="#">x86</a>   <a href="#">Arm64</a>   <a href="#">winget 指示</a> | <a href="#">x64</a>   <a href="#">x86</a>   <a href="#">Arm64</a>  |
| 全部      | <a href="#">dotnet-install scripts</a>  |  |

包含的執行階段

.NET 執行階段 8.0.23

ASP.NET Core 執行階段 8.0.23

.NET 桌面執行階段 8.0.23

語言支援

C# 12.0

F# 8.0

Visual Basic 16.9

➤ 點擊執行安裝檔

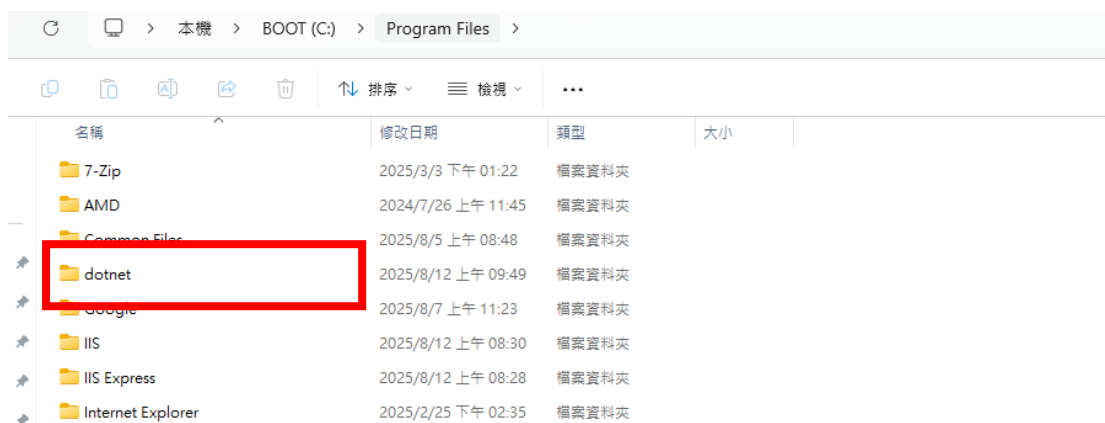


dotnet-sdk-8.0.417-win-x64.exe

210 MB • 22 分鐘前



➤ 至選擇的安裝路徑確認是否有 dotnet 資料夾



File Explorer window showing the path: 本機 > BOOT (C:) > Program Files.

| 名稱                | 修改日期               | 類型    | 大小 |
|-------------------|--------------------|-------|----|
| 7-Zip             | 2025/3/3 下午 01:22  | 檔案資料夾 |    |
| AMD               | 2024/7/26 上午 11:45 | 檔案資料夾 |    |
| Common Files      | 2025/8/5 上午 08:48  | 檔案資料夾 |    |
| dotnet            | 2025/8/12 上午 09:49 | 檔案資料夾 |    |
| Google            | 2025/8/7 上午 11:23  | 檔案資料夾 |    |
| IIS               | 2025/8/12 上午 08:30 | 檔案資料夾 |    |
| IIS Express       | 2025/8/12 上午 08:28 | 檔案資料夾 |    |
| Internet Explorer | 2025/2/25 下午 02:35 | 檔案資料夾 |    |

➤ 確認 dotnet 的版本是否有安裝正確

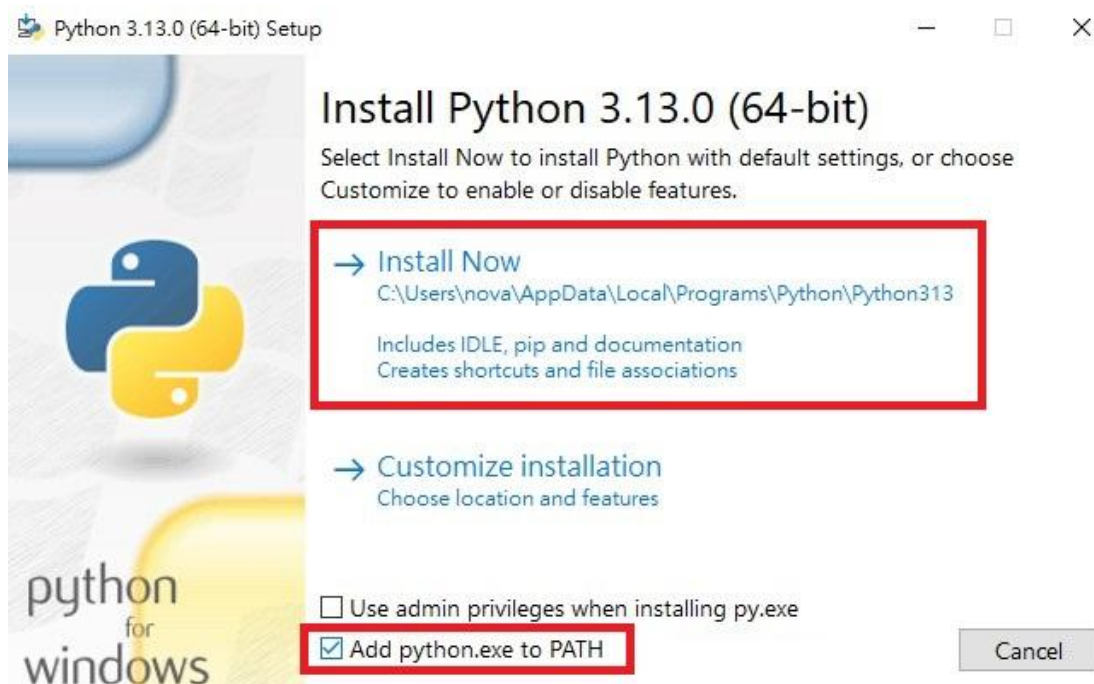
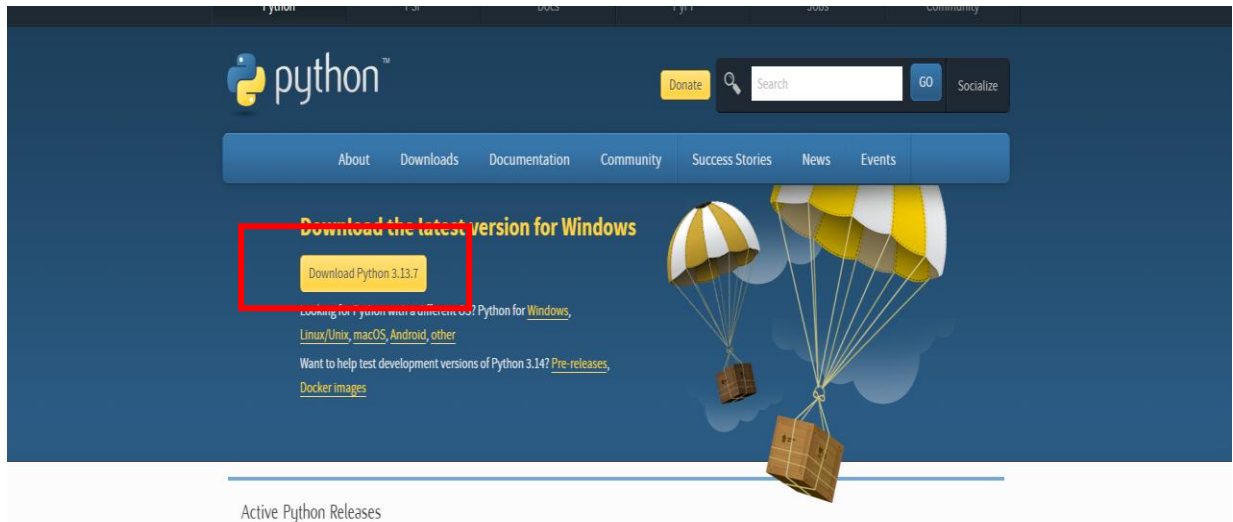


File Explorer window showing the path: 本機 > BOOT (C:) > Program Files > dotnet > sdk.

| 名稱      | 修改日期              | 類型    | 大小 |
|---------|-------------------|-------|----|
| 8.0.417 | 2026/2/4 下午 03:21 | 檔案資料夾 |    |

## 2. 安裝 Python 檔(若已安裝可省略)

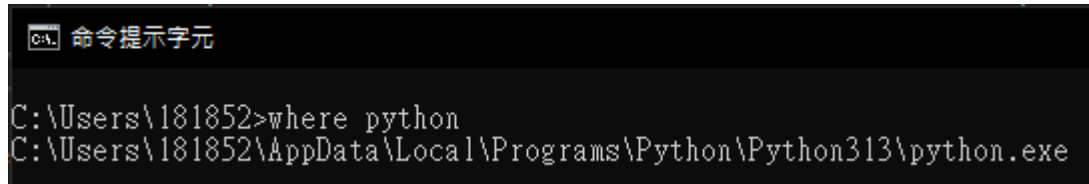
➤ 至官網下載 Python <https://www.python.org/downloads/>



- 若下載時未勾選 Add python.exe to PATH 的話，也可以手動將路徑

加入 PATH 內

1. 確認安裝位置→CMD 執行 `where python`

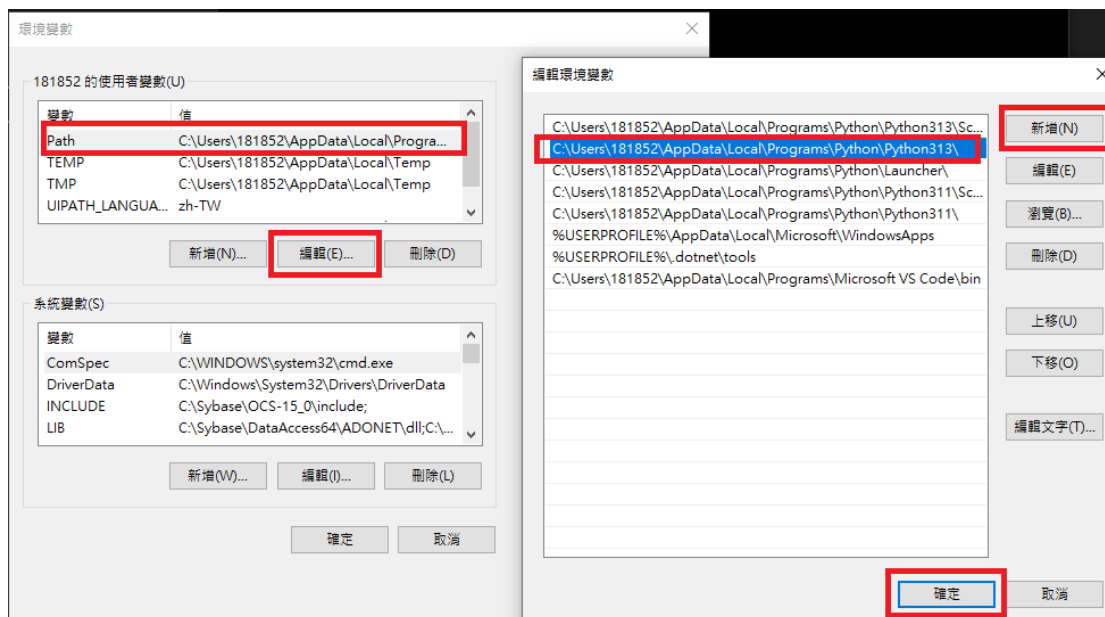


```
命令提示字元
C:\Users\181852>where python
C:\Users\181852\AppData\Local\Programs\Python\Python313\python.exe
```

2. 新增安裝路徑至 PATH

1. 開啟控制台：在搜尋列輸入「控制台」，然後選擇「控制台」應用程式。
2. 進入系統設定：在控制台中，找到並點擊「系統及安全性」，然後點擊「系統」。
3. 開啟進階系統設定：在系統視窗的左側，點擊「進階系統設定」。
4. 開啟環境變數：在「系統內容」視窗中，點擊「環境變數」按鈕。
5. 編輯PATH 變數：在「環境變數」視窗中，找到「系統變數」區段，然後找到名為「Path」的變數，點擊它，然後點擊「編輯」。
6. 修改PATH：在「編輯環境變數」視窗中，您可以添加、修改或刪除PATH 中的路徑。每個路徑之間用分號(;) 分隔。
7. 儲存變更：完成修改後，點擊「確定」關閉所有開啟的視窗。





### 3. pip 安裝 pythonnet 套件

- 確認 Python 環境是否安裝 pip
- 執行命令提示字元，輸入 `pip --version`
- 未安裝請參考官網說明 <https://pip.pypa.io/en/stable/installation>

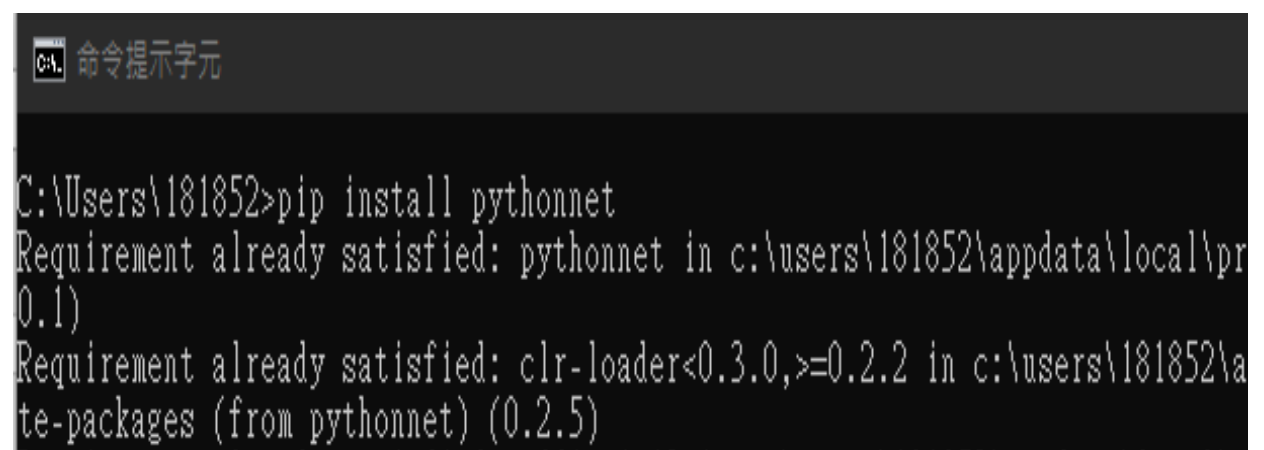


```
C:\> 命令提示字元

Microsoft Windows [版本 10.0.19044.5371]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\181852>pip --version
pip 23.1.2 from C:\Users\181852\AppData\Roaming\Python\Python38\site-packages\
```

- 安裝 Pythonnet 套件
- 執行命令提示字元，輸入 `pip install pythonnet`



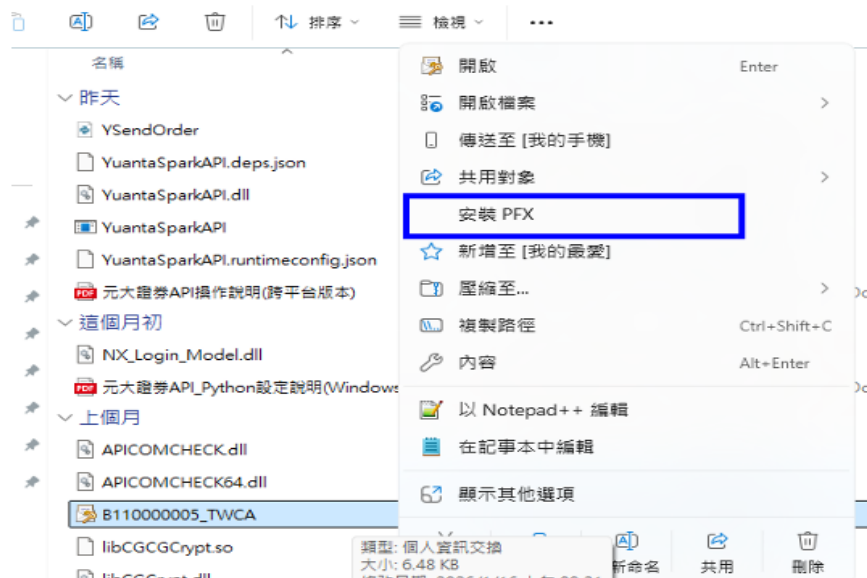
```
C:\> 命令提示字元

C:\Users\181852>pip install pythonnet
Requirement already satisfied: pythonnet in c:\users\181852\appdata\local\pr
(0.1)
Requirement already satisfied: clr-loader<0.3.0,>=0.2.2 in c:\users\181852\appdata\local\pr
te-packages (from pythonnet) (0.2.5)
```

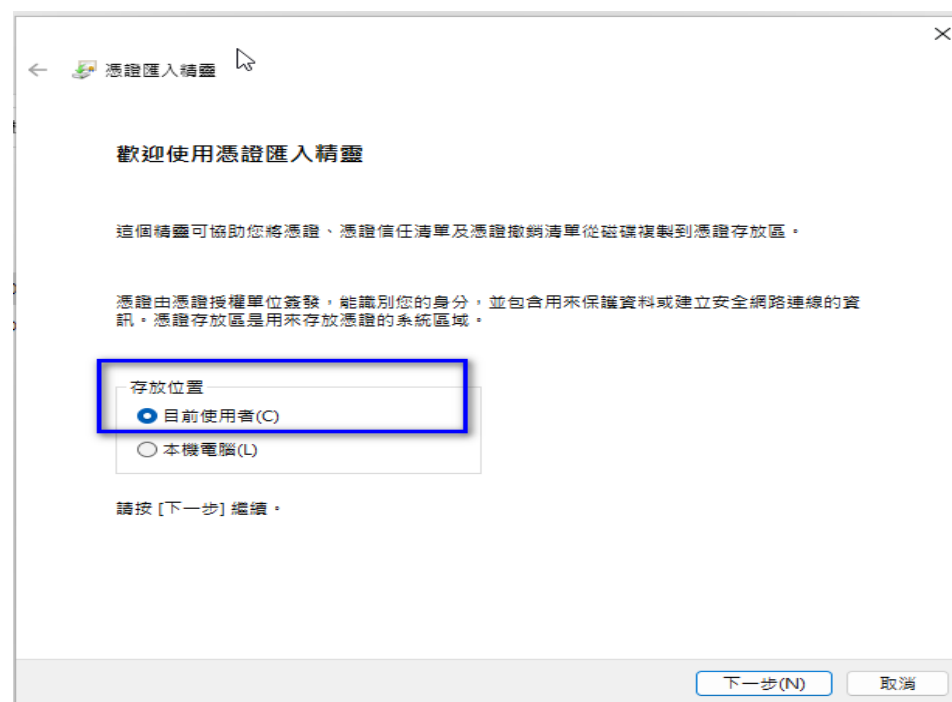
### 三、範例程式操作說明

#### (一)Windows

##### 1. 測試環境憑證匯入



➤ 安裝測試憑證在目前的帳號上



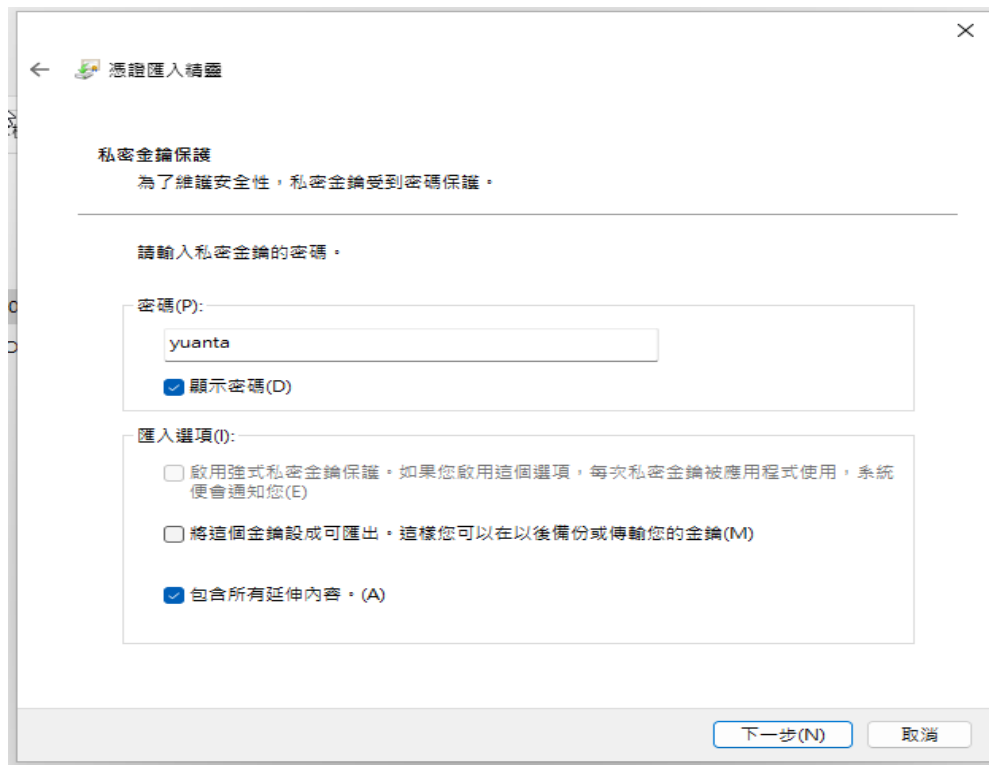
- 開放系統允許安裝更新憑證



- 安裝更新憑證



➤ 憑證安裝密碼：yuanta



← 憑證匯入精靈

**私密金鑰保護**  
為了維護安全性，私密金鑰受到密碼保護。

請輸入私密金鑰的密碼。

密碼(P):  
yuanta

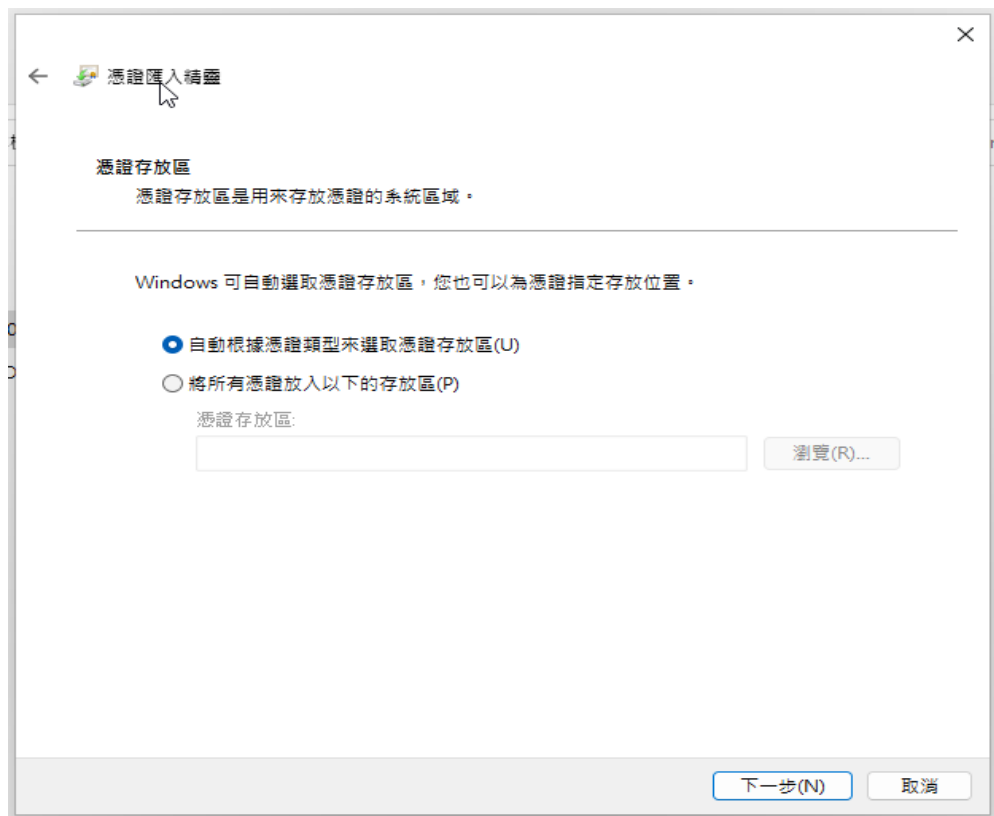
☒ 顯示密碼(D)

匯入選項(I):

- ☐ 啟用強式私密金鑰保護。如果您啟用這個選項，每次私密金鑰被應用程式使用，系統便會通知您(E)
- ☐ 將這個金鑰設成可匯出，這樣您可以在以後備份或傳輸您的金鑰(M)
- ☒ 包含所有延伸內容。(A)

下一步(N) 取消

➤ 憑證安裝作業



← 憑證匯入精靈

**憑證存放區**  
憑證存放區是用來存放憑證的系統區域。

Windows 可自動選取憑證存放區，您也可以為憑證指定存放位置。

☒ 自動根據憑證類型來選取憑證存放區(U)

☐ 將所有憑證放入以下的存放區(P)

憑證存放區:

瀏覽(R)...

下一步(N) 取消

➤ 完成憑證安裝作業



## 2. 主程式 YSendOrder.py

- 編輯與確認範例程式(元件引用路徑與範例程式路徑相同)

```
import os
#import clr_loader
import time
import datetime
import struct
import pathlib
import sys
from datetime import datetime
from pathlib import Path
from pythonnet import load

# 指定 Python.Runtime.dll 的路徑
#os.environ["PYTHONNET_PYDLL"] = r"C:\Users\*\AppData\Local\Programs\Python\Python38\python38.dll"

# 使用 clr_loader 載入 .NET Core/.NET 8
load("coreclr")

#其他載入寫法
#from clr_loader import get_coreclr
#from pythonnet import set_runtime
#rt = get_coreclr()
#set_runtime(rt)
#export PYTHONNET_RUNTIME=coreclr

import clr
import System
##透過Clr引用系統標準函式
clr.AddReference('System.Collections')
from System.Collections.Generic import List

##宣告增加模組、DLL的路徑(windows可抓取當前路徑 Linux跟MAC需指定路徑)
sys.path.append(Path(pathlib.Path(__file__).parent.resolve()))
if sys.platform == "win32":
    os.add_dll_directory(Path(pathlib.Path(__file__).parent.resolve()))

##透過Clr引用YuantaSparkAPI.dll
##pythonnet引用元件不用加附檔名
try:
    clr.AddReference("YuantaSparkAPI")
except Exception as e:
    print(f"Error loading YuantaSparkAPI: {e}")
```

- 使用 load 函數載入.NET Core/8 的執行環境

```
load("coreclr")
```

- 讓 Python 可以使用.NET 類別，例如 System.Collections.Generic.List，並設定 DLL 路徑，確保 Python 可以找到同資料夾下的 YuantaSparkAPI.dll

```
import clr
```

```
import System
```

```
clr.AddReference('System.Collections')
```

```
from System.Collections.Generic import List
```

```
sys.path.append(Path(pathlib.Path(__file__).parent.resolve()))
```

```
os.add_dll_directory(Path(pathlib.Path(__file__).parent.resolve()))
```

- 嘗試載入 YuantaSparkAPI.dll

```
try:
```

```
    clr.AddReference("YuantaSparkAPI")
```

```
except Exception as e:
```

```
    print(f"Error loading YuantaSparkAPI: {e}")
```



➤ 點擊執行範例程式

即可執行連線、登入、下單、收到即時回報資訊

請確保所有元件與主程式(YSendOrder.py)位於同一目錄或指定路徑

```
PS C:\Users\183385\Desktop\old_function\YuantaOneAPI_NET6_Python> py YSendOrder.py
#####

交易主機Is Connected!!

#####

00001,委託成功,帳號筆數:1
S98875005091,陳oo,8110000005,55

#####

台股報價/國內期貨報價/國外期貨報價Is Connected!!
#####

現貨下單結果:
0001,執行成功,下單筆數:1
1,0,f0990,2025/8/14,,,

#####

即時回報:
S98875005091,50,f0990,TWSE,2330,台積電,2025/8/14,13:9:16.447,0,8,1200.0,0.0,0,0,1000,,0,,1,0,,0,0,0,1,100026,0,2,00000

#####

即時回報彙總:
S98875005091,1,f0990,TWSE,2330,,2025/8/14,13:9:16.447,0,8,1200.0,0.0,0,0,0,0,1000,0,,,0,T,0,20,0,台積電,2330,,0.0,,0,0,1,100026,2,

#####

台股報價/國內期貨報價/國外期貨報價Is Disconnected
#####

交易主機Is Disconnected
```

➤ 程式下單部分為 send\_stock\_order()，可更換股票代碼以及下單股

數，詳細內容可查看 SendStockOrder\_IO\_Spec.doc 文件

```
958
959 #SendStockOrder
960 def send_stock_order(yuanta):
961     stockorder = StockOrder()
962     stockorder.Identify = 1
963     stockorder.Account = 'S98875005091'
964     stockorder.APCode = 0
965     stockorder.TradeKind = 0
966     stockorder.OrderType = '0'
967     stockorder.StkCode = '2330'
968     stockorder.PriceFlag = ''
969     stockorder.Price = 1200
970     stockorder.OrderQty = 1
971     stockorder.BuySell = 'B'
972     stockorder.SellerNo = 0
973     stockorder.OrderNo = ''
974     stockorder.TradeDate = datetime.today().strftime('%Y/%m/%d')
975     stockorder.BasketNo = ''
976     stockorder.Time_in_force = '0'
977
978     lstStockOrder = List[StockOrder]()
979     lstStockOrder.Add(stockorder)
980
981     #傳送下單
982     yuanta.SendStockOrder('S98875005091', lstStockOrder)
983     #測試環境傳送後要休息一下
984     time.sleep(1)
985
```

➤ 報價表部分為 ReadWatchListAll\_api()，提供股票價格等當前資訊，

可自行增加股票名單，詳細內容可查看 GetWatchList\_IO\_Spec.doc

文件

```
#GetWatchListAll
def ReadWatchListAll_api(yuanta):
    quoteList = List[Quote]()
    quote = Quote()
    quote.MarketType = enumMarketType.TWSE
    quote.StockCode = '2885'
    quoteList.Add(quote)
    yuanta.GetWatchListAll('S98875005091', quoteList)
```

## (二)MacOS

### 主程式 YSendOrder.py

#### ➤ 編輯與確認範例程式

```
import os
import clr_loader
import time
import datetime
import struct
import pathlib
import sys
from datetime import datetime
from pathlib import Path
from pythonnet import load

# 指定 Python.Runtime.dll 的路徑
#os.environ["PYTHONNET_PYDLL"] = r"C:\Users\*\AppData\Local\Programs\Python\Python38\python38.dll"

# 使用 clr_loader 載入 .NET Core/.NET 8
load("coreclr")

#其他載入寫法
#from clr_loader import get_coreclr
#from pythonnet import set_runtime
#rt = get_coreclr()
#set_runtime(rt)
#export PYTHONNET_RUNTIME=coreclr

import clr
import System
##透過Clr引用系統標準函式
clr.AddReference('System.Collections')
from System.Collections.Generic import List

##宣告增加模組、DLL的路徑(windows可抓取當前路徑 Linux跟MAC需指定路徑)
sys.path.append(Path(pathlib.Path(__file__).parent.resolve()))
if sys.platform == "win32":
    os.add_dll_directory(Path(pathlib.Path(__file__).parent.resolve()))

##透過Clr引用YuantaSparkAPI.dll
##pythonnet引用元件不用加附檔名
try:
    clr.AddReference("YuantaSparkAPI")
except Exception as e:
    print(f"Error loading YuantaSparkAPI: {e}")
```

- 使用 load 函數載入.NET Core/8 的執行環境

```
load("coreclr")
```

- 讓 Python 可以使用.NET 類別，例如 System.Collections.Generic.List，並

設定 DLL 路徑(MAC 需指定路徑)，確保可以找到 YuantaSparkAPI.dll

```
import clr
```

```
import System
```

```
clr.AddReference('System.Collections')
```

```
from System.Collections.Generic import List
```

```
os.add_dll_directory(指定的絕對路徑)
```

- 嘗試載入 YuantaSparkAPI.dll

```
try:
```

```
    clr.AddReference("YuantaSparkAPI")
```

```
except Exception as e:
```

```
    print(f"Error loading YuantaSparkAPI: {e}")
```

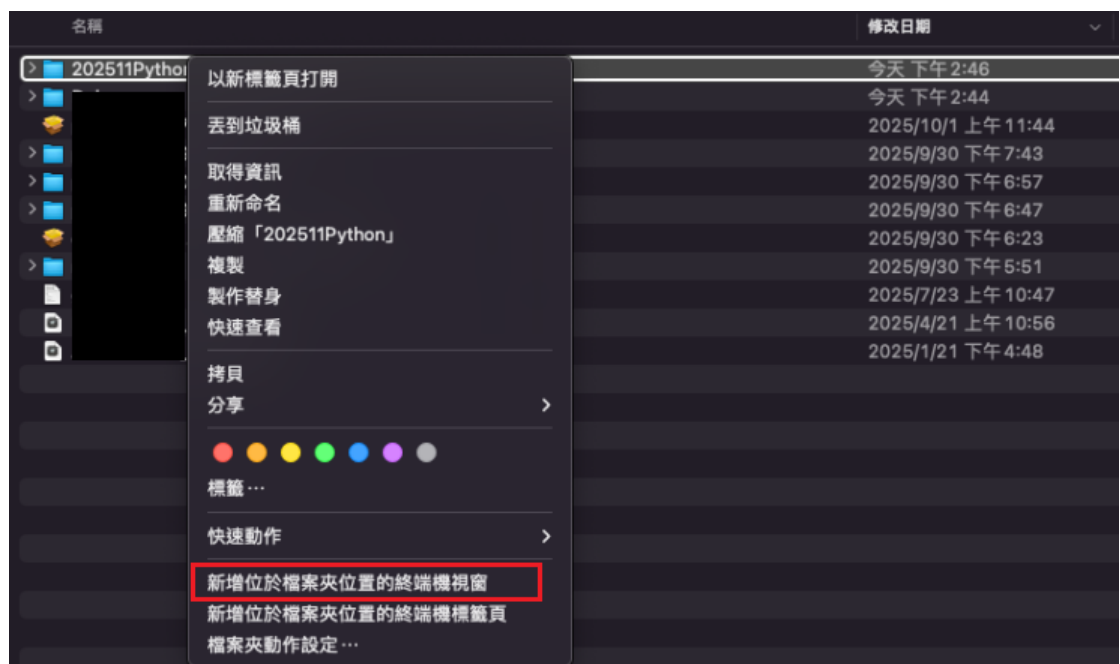
- 調整登入寫法，MacOS 需指定讀取的憑證才能進行登入，路徑請填

寫絕對路徑，其他欄位說明可查看 Login\_IO\_Spec.doc 文件

```
#Open
def open_api(yuanta):
    yuanta.Open(enumEnvironmentMode.UAT)
    #time.sleep(3)

#Login
def login_api(yuanta):
    #現貨
    #Windows登入
    #yuanta.Login('S98875005091', '1234')
    #期貨
    #yuanta.Login('FF0210132243219588', 'abcd123')
    #yuanta.Login('FF0210006702606001', 'abcd123')
    #Linux登入 需帶入憑證絕對路徑與憑證密碼
    yuanta.Login('/Users/ycpsts/MACTEST/B110000005_TWCA.pfx', 'yuanta', 'S98875005091', '1234')
    #yuanta.Login('/Users/ycpsts/MACTEST/A110000004_TWCA.pfx', 'yuanta', 'FF021919F000168885', 'abcd123')
    #time.sleep(3)
```

- 於範例程式資料夾開啟終端機畫面



➤ 語法執行範例程式

MacOS 環境的 python 執行需輸入 python 版本(python3)

即可執行連線、登入、下單、收到即時回報資訊

請確保所有元件與主程式(YSendOrder.py)位於同一目錄或指定路徑

```
202511Python — Python YSendOrder.py — 85x40
Last login: Mon Oct 27 14:12:19 on ttys002
[ycpsts@Y220003079-MAC 202511Python % python3 YSendOrder.py
##=====##

交易主機 Is Connected!!

##=====##

00001,委託成功,帳號筆數:1
S98875005091,陳○○,B110000005,55

##=====##

台股報價/國內期貨報價/國外期貨報價 Is Connected!!

##=====##

現貨下單結果:
0001,執行成功,下單筆數:1
1,0,f0045,2025/10/27,,,

##=====##

即時回報彙總:
S98875005091,1,f0045,TWSE,2885,2025/10/27,15:1:33.568,0,8,35.15,0.0,0
.0,0.0,0,1000,0,,,0,T,0,20,0,元大金,2885,0.0,0,0,1
,100021,2,

##=====##

即時回報:
S98875005091,50,f0045,TWSE,2885,元大金,2025/10/27,15:1:33.568,0,8,35.15,0.0,0,1000,
,0,,1,0,,0,0,0,1,100021,0,2,00000
```

- 程式下單部分為 `send_stock_order()`，可更換股票代碼以及下單股數，詳細內容可查看 `SendStockOrder_IO_Spec.doc` 文件

```
#SendStockOrder
def send_stock_order(yuanta):
    stockorder = StockOrder()
    stockorder.Identify = 1
    stockorder.Account = 'S98875005091'
    stockorder.APCode = 0
    stockorder.TradeKind = 0
    stockorder.OrderType = '0'
    stockorder.StkCode = '2885'
    stockorder.PriceFlag = ''
    stockorder.Price = 35.15
    stockorder.OrderQty = 1
    stockorder.BuySell = 'B'
    stockorder.OrderNo = ''
    stockorder.TradeDate = datetime.today().strftime('%Y/%m/%d')
    stockorder.BasketNo = ''
    stockorder.Time_in_force = '0'

    lstStockOrder = List[StockOrder]()
    lstStockOrder.Add(stockorder)

    #傳送下單
    yuanta.SendStockOrder('S98875005091', lstStockOrder)
    #測試環境傳送後要休息一下
    time.sleep(1)
```

- 報價表部分為 `ReadWatchListAll_api()`，提供股票價格等當前資訊，可自行增加股票名單，詳細內容可查看 `GetWatchList_IO_Spec.doc` 文件

```
#GetWatchListAll
def ReadWatchListAll_api(yuanta):
    quoteList = List[Quote]()
    quote = Quote()
    quote.MarketType = enumMarketType.TWSE
    quote.StockCode = '2885'
    quoteList.Add(quote)
    yuanta.GetWatchListAll('S98875005091', quoteList)
```

## 四、備註

### 1. 可能會出現的錯誤

- 未安裝.NET SDK

```
Traceback (most recent call last):
  File "C:\Users\183385\AppData\Local\Programs\Python\Python313\Lib\site-packages\pythonnet\_init_.py", line 77, in _create_runtime_from_spec
    return clr_loader.get_coreclr(**params)
           ~~~~~^~~~~~
  File "C:\Users\183385\AppData\Local\Programs\Python\Python313\Lib\site-packages\clr_loader\_init_.py", line 121, in get_coreclr
    dotnet_root = find_dotnet_root()
  File "C:\Users\183385\AppData\Local\Programs\Python\Python313\Lib\site-packages\clr_loader\util\find.py", line 57, in find_dotnet_root
    raise RuntimeError("Can not determine dotnet root")
RuntimeError: Can not determine dotnet root
```

The above exception was the direct cause of the following exception:

```
Traceback (most recent call last):
  File "C:\Users\183385\Desktop\old_function\YuantaOneAPI_NET6_Python\YSendOrder.py", line 21, in <module>
    load("coreclr")
    ~~~~^~~~~~
  File "C:\Users\183385\AppData\Local\Programs\Python\Python313\Lib\site-packages\pythonnet\_init_.py", line 135, in load
    set_runtime(runtime, **params)
    ~~~~~^~~~~~
  File "C:\Users\183385\AppData\Local\Programs\Python\Python313\Lib\site-packages\pythonnet\_init_.py", line 29, in set_runtime
    runtime = _create_runtime_from_spec(runtime, params)
  File "C:\Users\183385\AppData\Local\Programs\Python\Python313\Lib\site-packages\pythonnet\_init_.py", line 90, in _create_runtime_from_spec
    raise RuntimeError(
    ...<2 lines>...
    ) from exc
RuntimeError: Failed to create a .NET runtime (coreclr) using the
parameters {}.
```

C:\Users\183385\Desktop\old\_function\YuantaOneAPI\_NET6\_Python\





- 這代表 Python 無法正確找到.NET，您可以將 C:\Program Files\dotnet 資料夾刪除並重新安裝.NET 8 以上的版本

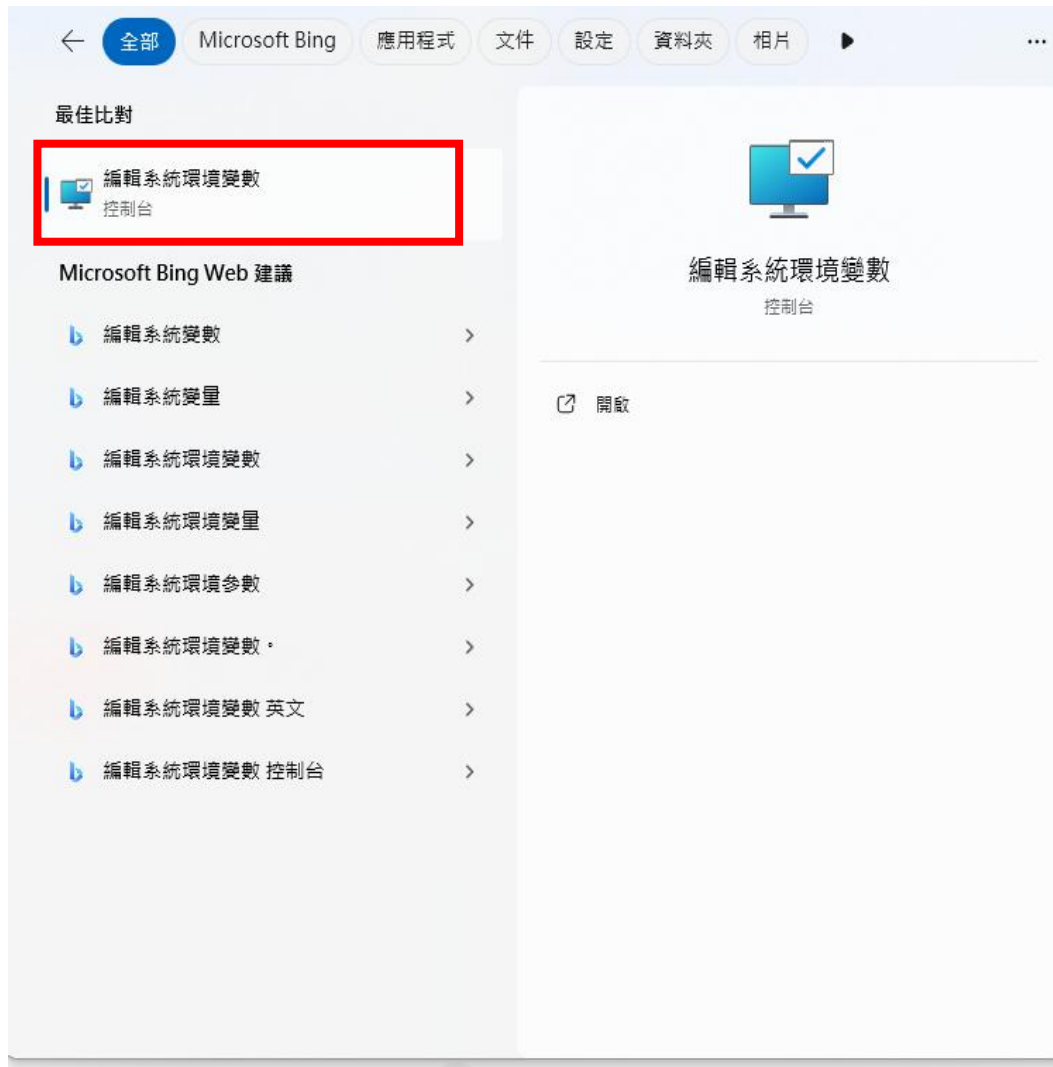
```
PS C:\Users\183385\Desktop\old_function\YuantaOneAPI_NET6_Python> py YSendOrder.py
A fatal error was encountered. The library 'hostpolicy.dll' required to execute the application was not found in 'C:\Program Files\dotnet\shared\Microsoft.N
Traceback (most recent call last):
  File "C:\Users\183385\AppData\Local\Programs\Python\Python313\Lib\site-packages\pythonnet\_init_.py", line 77, in _create_runtime_from_spec
    return clr_loader.get_coreclr(**params)
           ~~~~~^~~~~~
  File "C:\Users\183385\AppData\Local\Programs\Python\Python313\Lib\site-packages\clr_loader\_init_.py", line 142, in get_coreclr
    impl = DotnetCoreRuntime(runtime_config=runtime_config, dotnet_root=dotnet_root)
  File "C:\Users\183385\AppData\Local\Programs\Python\Python313\Lib\site-packages\clr_loader\hostfxr.py", line 23, in __init__
    self.handle = _get_handle(self.dll, self.dotnet_root, runtime_config)
                  ~~~~~^~~~~~
  File "C:\Users\183385\AppData\Local\Programs\Python\Python313\Lib\site-packages\clr_loader\hostfxr.py", line 138, in _get_handle
    check_result(res)
    ~~~~~^~~~~~
  File "C:\Users\183385\AppData\Local\Programs\Python\Python313\Lib\site-packages\clr_loader\util\_init_.py", line 42, in check_result
    raise error
clr_loader.util.clr_error.ClrError: 0x80008083: CoreHostLibMissingFailure

The above exception was the direct cause of the following exception:

Traceback (most recent call last):
  File "C:\Users\183385\Desktop\old_function\YuantaOneAPI_NET6_Python\YSendOrder.py", line 21, in <module>
    load("coreclr")
    ~~~~~^~~~~~
  File "C:\Users\183385\AppData\Local\Programs\Python\Python313\Lib\site-packages\pythonnet\_init_.py", line 135, in load
    set_runtime(runtime, **params)
    ~~~~~^~~~~~
  File "C:\Users\183385\AppData\Local\Programs\Python\Python313\Lib\site-packages\pythonnet\_init_.py", line 29, in set_runtime
    runtime = _create_runtime_from_spec(runtime, params)
  File "C:\Users\183385\AppData\Local\Programs\Python\Python313\Lib\site-packages\pythonnet\_init_.py", line 90, in _create_runtime_from_spec
    raise RuntimeError(
    ...<2 lines>...
    ) from exc
RuntimeError: Failed to create a .NET runtime (coreclr) using the
parameters {}.
```

➤ 若還是無法解決問題可以在搜尋列中搜尋編輯系統環境變數中加

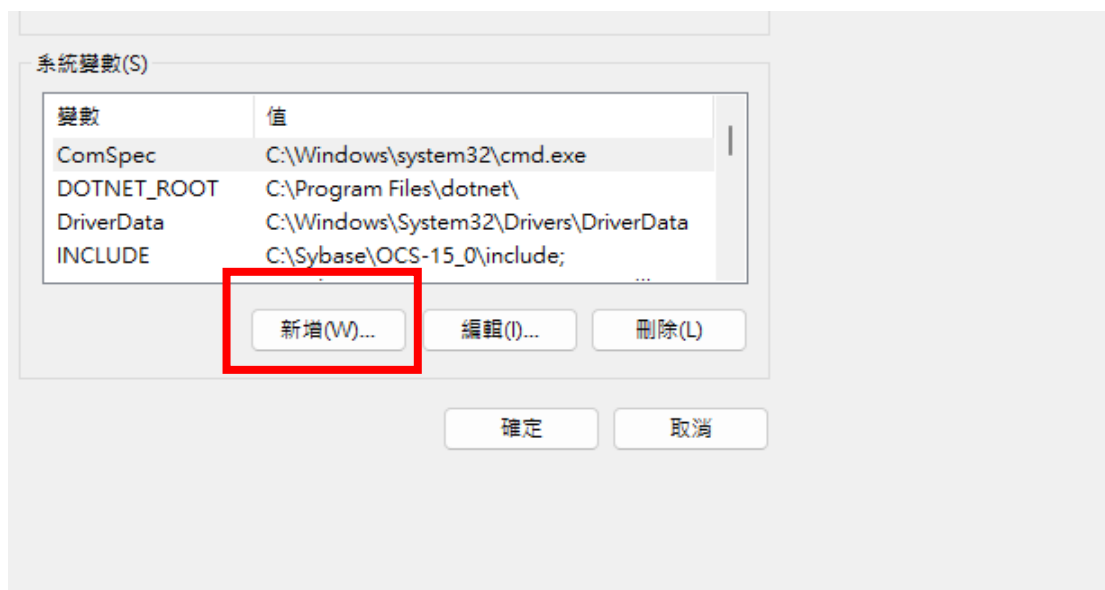
入:DOTNET\_ROOT=C:\Program Files\dotnet\



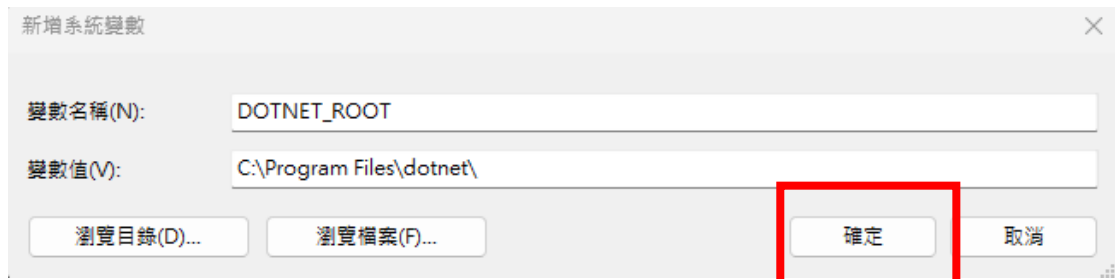
➤ 點擊環境變數



➤ 新增系統變數



- 加入:DOTNET\_ROOT=C:\Program Files\dotnet\，點擊確定後重新執行程式



## 2. 方法參數型別

Python 語言的數字型別為 int (整數)、float (浮點數)，故使用各項功能傳遞參

數時，若非文件說明為浮點數之型別，請以 int 型別傳遞，例如現貨下單：

| Function   | bool SendStockOrder(string , List<StockOrder> ,<br>enumLangType) |        |      |                                   |
|------------|--|--------|------|-----------------------------------|
| Field Name | Description  | Type   | Size | Memo                              |
| StockOrder | 下單物件   | Class  |      |                                   |
| TradeKind  | 交易性質   | short  |      | 00:委託單<br>03:改量<br>04:取消<br>07:改價 |
| Price      | 委託價格   | double |      | 非限價請填 0                           |

```
#SendStockOrder
def send_stock_order(yuanta):
    stockorder = StockOrder()
    stockorder.Identify = 1
    stockorder.Account = 'S98875005091'
    stockorder.APCode = 0
    stockorder.TradeKind = 0
    stockorder.OrderType = '0'
    stockorder.StkCode = '2330'
    stockorder.PriceFlag = ''
    stockorder.Price = 1200
    stockorder.OrderQty = 1
    stockorder.BuySell = 'B'
    stockorder.SellerNo = 0
    stockorder.OrderNo = ''
    stockorder.TradeDate = datetime.today().strftime('%Y/%m/%d')
    stockorder.BasketNo = ''
    stockorder.Time_in_force = '0'

    lstStockOrder = List[StockOrder]()
    lstStockOrder.Add(stockorder)

    #傳送下單
    yuanta.SendStockOrder('S98875005091', lstStockOrder)
    #測試環境傳送後要休息一下
    time.sleep(1)
```