**群益Python API支援3大功能**

**報價**

**歷史K**

**下單**

**回報**

以下每一個步驟解說，皆包含簡短程式碼，最後附上完整程式碼讓你可以直接複製貼上，即可開始使用，一點都不麻煩。

下面是解釋透過Python開發使用群益API時，需要了解有那些程序：

1.初始化群益COM元件-工欲善其事，必先利其器。

2.使用Python建立windows視窗-規劃一個美美的個人化視窗吧。

3.啟動報價/抓歷史K-行情波動閃爍真迷人,心跳開始加速了嗎。

4.串接下單功能-配合報價資料，就可以來下單囉。

5.成果回報-立即來檢視一下獲利吧。

***初始化群益元件***

**群益API COM元件由群益金融網下載，內含說明文件－教學你如何註冊。使用前及更新版本，需先註冊COM元件。**

群益API COM底層是由C++編寫而成，有強大運算能力且擁有速度上的優勢，而使用不同的程式語言必須要透過相關引用([安裝](#環境設定))，才能在Python範例程式碼直接介接群益API COM元件，以下為安裝套件後，介接x64 COM範例：

# 先把API com元件初始化

import comtypes.client

comtypes.client.GetModule(r'./x64/SKCOM.dll')

import comtypes.gen.SKCOMLib as sk

# 第一種讓群益API元件可導入讓Python code使用的方法

import win32com.client

from ctypes import WinDLL,byref

from ctypes.wintypes import MSG

SKCenterLib = win32com.client.Dispatch("{AC30BAB5-194A-4515-A8D3-6260749F8577}")

SKOrderLib = win32com.client.Dispatch("{54FE0E28-89B6-43A7-9F07-BE988BB40299}")

# 第二種讓群益API元件可導入Python code內用的物件宣告

skC = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)

skO = comtypes.client.CreateObject(sk.SKOrderLib,interface=sk.ISKOrderLib)

skQ = comtypes.client.CreateObject(sk.SKQuoteLib,interface=sk.ISKQuoteLib)

skR = comtypes.client.CreateObject(sk.SKReplyLib,interface=sk.ISKReplyLib)

# 畫視窗用物件

from tkinter import \*

from tkinter.ttk import \*

from tkinter import messagebox,colorchooser,font,Button,Frame,Label

# 數學計算用物件

import math

# 顯示各功能狀態用的function

def WriteMessage(strMsg,listInformation):

listInformation.insert('end', strMsg)

listInformation.see('end')

def SendReturnMessage(strType, nCode, strMessage,listInformation):

GetMessage(strType, nCode, strMessage,listInformation)

def GetMessage(strType,nCode,strMessage,listInformation):

strInfo = ""

if (nCode != 0):

strInfo ="【"+ skC.SKCenterLib\_GetLastLogInfo()+ "】"

WriteMessage("【" + strType + "】【" + strMessage + "】【" + skC.SKCenterLib\_GetReturnCodeMessage(nCode) + "】" + strInfo,listInformation)

以上程式碼的部分為初始化元件與主要相關功能宣告，未初始化是無法讓程式順利運作的，以上初始化的程式，不管在報價、下單或是回報都會用到，看不懂沒關係，別給自己太大壓力，[有完整程式](#報價)範例與環境設定，直接打開複製貼上或直接執行範例。

***使用Python建立windows視窗***

大部分Python程式寫出來都是console畫面，為了要讓各位看官方便使用，我們開發了一個視窗介面的範例，讓操作更容易，顯示資料較直接等等。

以下僅列出部分UI介面設計相關的程式碼，方便大家參考。

#上半部登入框

class FrameLogin(Frame):

def \_\_init\_\_(self, master = None):

Frame.\_\_init\_\_(self, master)

self.grid()

#self.pack()

self.place()

self.FrameLogin = Frame(self)

self.master["background"] = "#ffecec"

self.FrameLogin.master["background"] = "#ffecec"

self.createWidgets()

def createWidgets(self):

#帳號

self.labelID = Label(self)

self.labelID["text"] = "帳號："

self.labelID["background"] = "#ffecec"

self.labelID["font"] = 20

self.labelID.grid(column=0,row=0)

#輸入框

self.textID = Entry(self)

self.textID["width"] = 50

self.textID.grid(column = 1, row = 0)

#密碼

self.labelPassword = Label(self)

self.labelPassword["text"] = "密碼："

self.labelPassword["background"] = "#ffecec"

self.labelPassword["font"] = 20

self.labelPassword.grid(column = 2, row = 0)

#輸入框

self.textPassword = Entry(self)

self.textPassword["width"] = 50

self.textPassword['show'] = '\*'

self.textPassword.grid(column = 3, row = 0)

#按鈕

self.buttonLogin = Button(self)

self.buttonLogin["text"] = "Login"

self.buttonLogin["background"] = "#ff9797"

self.buttonLogin["foreground"] = "#000000"

self.buttonLogin["highlightbackground"] = "#ff0000"

self.buttonLogin["font"] = 20

self.buttonLogin["command"] = self.buttonLogin\_Click

self.buttonLogin.grid(column = 4, row = 0)

#ID

self.labelID = Label(self)

self.labelID["text"] = "<<ID>>"

self.labelID["background"] = "#ffecec"

self.labelID["font"] = 20

self.labelID.grid(column = 5, row = 0)

#訊息欄

self.listInformation = Listbox(root, height=5)

self.listInformation.grid(column = 0, row = 1, sticky = E + W)

global GlobalListInformation,Global\_ID

GlobalListInformation = self.listInformation

Global\_ID = self.labelID

# 這裡是登入按鈕,使用群益API不管要幹嘛你都要先登入才行

def buttonLogin\_Click(self):

try:

m\_nCode = skC.SKCenterLib\_Login(self.textID.get().replace(' ',''),self.textPassword.get().replace(' ',''))

if(m\_nCode==0):

Global\_ID["text"] = self.textID.get().replace(' ','')

WriteMessage("登入成功",self.listInformation)

else:

WriteMessage(m\_nCode,self.listInformation)

except Exception as e:

messagebox.showerror("error！",e)

上面黃色框線部分，是很重要的登入功能，黃色字體部份是最關鍵的部分－登入函式，其他都只是畫面顯示用的設定，看不懂沒關係，別給自己太大壓力，[有完整程式範例](#報價)與環境設定，直接打開複製貼上或直接執行範例。

***啟動報價***

取得報價資訊分為兩個步驟：

1. 呼叫一個叫SKQuoteLib\_EnterMonitor()的函式，這部分是連接群益報價主機，在範例程式大概是下面片段：

def btnConnect\_Click(self):

try:

m\_nCode = skQ.SKQuoteLib\_EnterMonitor()

SendReturnMessage("Quote", m\_nCode, "startQuote",GlobalListInformation)

except Exception as e:

messagebox.showerror("error！",e)

2. 接下來，想要什麼股票的報價，則必須透過SKQuoteLib\_RequestStocks函式送出商品的代號給API COM元件：

….

def btnQueryStocks\_Click(self):

try:

if(self.txtPageNo.get().replace(' ','') == ''):

pn = 0

else:

pn = int(self.txtPageNo.get())

skQ.SKQuoteLib\_RequestStocks(pn,self.txtStocks.get().replace(' ',''))

except Exception as e:

messagebox.showerror("error！",e)

….

由於在前端我們會提供一個輸入框讓你輸入一串(一個以上)股票代號self.txtStocks.get().replace(' ','') 代表是從輸入端取得的股票代碼，例如：2330，其中pn的作用說明如下，由於群益API可在同一時間提供非常多商品報價，非常非常多，為了有效處理這麼多檔商品的報價資料進而方便使用者分類顯示，類似頁簽的概念，在自己寫的程式裡可以將報價分別放在不同分頁；假設以產業分類，在第一頁想看食品類20檔報價，第二頁為光電類50檔，這個pn你可以填1跟2，但初學者可以填1。

ex: skQ.SKQuoteLib\_RequestStocks(1, "2330") # 這樣就完成跟群益要台積電報價的程序了

3. 註冊完報價呢? 已註冊的股票報價資料會從下面片段程式，進行處理：

……

def OnNotifyQuote(self, sMarketNo, sStockidx):

pStock = sk.SKSTOCK()

skQ.SKQuoteLib\_GetStockByIndex(sMarketNo, sStockidx, pStock)

strMsg = '代碼:',pStock.bstrStockNo,'--名稱:',pStock.bstrStockName,'--開盤價:',pStock.nOpen/math.pow(10,pStock.sDecimal),'--最高:',pStock.nHigh/math.pow(10,pStock.sDecimal),'--最低:',pStock.nLow/math.pow(10,pStock.sDecimal),'--成交價:',pStock.nClose/math.pow(10,pStock.sDecimal),'--總量:',pStock.nTQty

WriteMessage(strMsg,Gobal\_Quote\_ListInformation)

……

使用者可以在OnNotifyQuote()加入中斷點，確認報價資料是如何由指定的事件送進來，這就是個callback function的概念，callback又是什麼概念，這大概像是你想看報紙，於是你到報社去訂報紙，你當然可以在報社等當天的報紙下來，然後每天去報社拿，但這樣太累，何不告訴報社你家住址讓報社每天送報紙給你呢? 宣告一個callback function就是告訴報社你家住址在哪，報社一有報紙(報價)就丟給你，你只需要到信箱收報紙(報價)即可。

另外解釋一下，SKQuoteLib\_GetStockByIndex是拿來做什麼的，由於OnNotifyQuote()由群益主機送下來的資訊只有群益用的商品編號跟市場編號，這兩個編號用途是取得SKSTOCKS完整的內容，群益報價主機為各市場及其商品自訂的索引。因此註冊報價後，要如何知道自己註冊的股票完整資訊，必須透過這個索引編號SKQuoteLib\_GetStockByIndex才能真正拿到該檔股票的詳細資訊。

群益也同時提供了歷史行情讓使用者計算、回測等等，相關的功能可從範例程式碼取得，很簡單，也容易上手。

看不懂沒關係，別給自己太大壓力，有完整程式範例與環境設定，直接打開複製貼上或直接執行範例。

***群益下單功能***

下單步驟比報價複雜一點，因為這是要去外面跟人家拚輸贏，所以要謹慎檢核送出的委託內容：

這邊是簡單介紹下單步驟，讓大家有印象。

首先介紹下單初始函式，下單需要先進行相關的設定：

#1.下單物件初始

def btnInitialize\_Click(self):

try:

m\_nCode = skO.SKOrderLib\_Initialize()

SendReturnMessage("Order", m\_nCode, "Order\_Initialize",GlobalListInformation)

except Exception as e:

messagebox.showerror("error！",e)

第二部分是檢查憑證，讀取憑證是為了讓使用者的交易更安全，確保本人交易：

#2.讀取憑證

def btnReadCert\_Click(self):

try:

m\_nCode = skO.ReadCertByID(Global\_ID["text"])

SendReturnMessage("Order", m\_nCode, "ReadCertByID",GlobalListInformation)

except Exception as e:

messagebox.showerror("error！",e)

第三部分是讀取下單帳號，你一定會記得你的身分證，但你可能不清楚你下單帳號是什麼，如果你開多個戶頭的話，則需要選擇交易帳號，下單時，這一定要代入的參數：

#3.取得下單帳號

def btnGetAccount\_Click(self):

try:

m\_nCode = SKOrderLib.GetUserAccount()

SendReturnMessage("Order", m\_nCode, "GetAccount",GlobalListInformation)

except Exception as e:

messagebox.showerror("error！",e)

呼叫GetUserAccount()之後的帳號從哪裡接收呢?這裡需要另一個callback function，來接收由群益主機傳送過來的帳戶資訊：

def OnAccount(self,bstrLogInID,bstrAccountData):

strValues = bstrAccountData.split(',')

strAccount = strValues[1] + strValues[3]

if strValues[0] == 'TS':

GlobalboxStockAccount['values'] = (strAccount)

上面這段code會接收你登入ID的所有帳號資訊，經過字串處理之後就可以拿來使用了。

第四部分就是下單-委託單內容相關的函式：

#4.下單送出

def btnSendStockOrder\_Click(self):

try:

if GlobalboxPrime.get()=="上市櫃":

sPrime=0

elif GlobalboxPrime.get()=="興櫃":

sPrime=1

if GlobalboxPeriod.get()=="盤中":

sPeriod=0

elif GlobalboxPeriod.get()=="盤後":

sPeriod=1

elif GlobalboxPeriod.get()=="零股":

sPeriod=2

if GlobalboxFlag.get()=="現股":

sFlag=0

elif GlobalboxFlag.get()=="融資":

sFlag=1

elif GlobalboxFlag.get()=="融券":

sFlag=2

elif GlobalboxFlag.get()=="無券":

sFlag=3

if GlobalboxBidAsk.get()=="買進":

sBuySell=0

elif GlobalboxBidAsk.get()=="賣出":

sBuySell=1

# 建立下單用的參數物件

oStock=sk.STOCKORDER()

# 填入完整帳號

oStock.bstrFullAccount = GlobalboxStockAccount.get()

# 填入股票代號

oStock.bstrStockNo = GlobaltxtStockNo.get()

# 上市、上櫃、興櫃

oStock.sPrime = sPrime

# 盤中、盤後、零股

oStock.sPeriod = sPeriod

# 現股、融資、融券

oStock.sFlag = sFlag

# 買賣別

oStock.sBuySell = sBuySell

# 委託價

oStock.bstrPrice = GlobaltxtPrice.get()

# 委託數量

oStock.nQty = int(GlobaltxtQty.get())

message,m\_nCode = skO.SendStockOrder(Global\_ID["text"], False, oStock)

SendReturnMessage("Order", m\_nCode, "GetUserAccount",GlobalListInformation)

except Exception as e:

messagebox.showerror("error！",e)

上面的程式碼，涵括了下單時要必須填入的所有參數，例如：股號，買賣別，現股還是融資、融券，盤中單、盤後單等等，黃色的部分是最後送出委託單的函式，執行完這行，代表委託單送到實際的股票市場進行條件的搓合；委託單送出前，需自行進行委託條件的檢查！看不懂沒關係，別給自己太大壓力，[[有完整程式範例](#報價)](#下單)與環境設定，直接打開複製貼上或直接執行範例。

***回報功能介紹***

下單後，一定要確認回報，尤其在程式交易裡面，要確認下單指令到底有沒有正確的執行，下面簡短的介紹回報功能：

def btnConnect\_Click(self):

try:

m\_nCode = skR.SKReplyLib\_ConnectByID("F123456789")

except Exception as e:

messagebox.showerror("error！",e)

上面的程式碼，是為了建立一個ID:F123456789與群益回報主機的連線，透過群益主機，把回報發送到使用者本機端；回報連線成功後，回報資料可以從下面的通知函式接收，這個callback function是接收登入ID所有回報資料：

def OnNewData(self,strUserID,StrData):

WriteMessage(strUserID + " " +StrData ,ReplyInformation)

回報功能除了上述的連線及接收資料，還有後續回報資料的處理，可參考策略王COM元件使用說明文件裡的回報欄位介紹，依你的設計需求,進行字串的處理。

看不懂沒關係，別給自己太大壓力，[這裡備有整份程式碼](#回報)與環境設定，直接打開複製貼上或直接執行就會動。

***總結***

上面我們提到最重要的function以下做一個總整理：

SKCenterLib\_Login() #使用群益API不管報價、下單、回報一定都要先登入

SKQuoteLib\_EnterMonitor() #使用報價前要先呼叫的function

SKQuoteLib\_RequestStocks() #告訴群益API你想要什麼股票的報價

OnNotifyQuote() #SKQuoteLib\_RequestStocks()的callback function

SKQuoteLib\_GetStockByIndex() #取得商品的詳細資訊

SKOrderLib\_Initialize() #下單要先初始的函式

ReadCertByID() #讀取憑證

GetUserAccount() #讀取帳戶基本資料

OnAccount() #GetUserAccount()的callback function

SendStockOrder() #下單送單函式

SKReplyLib\_ConnectByID() #連接群益回報主機

OnNewData() #SKReplyLib\_ConnectByID() 的callback function

以上Function都可以在 策略王COM元件使用說明XXXX.doc 文件裡面，找到相關功能及詳細參數說明，文件附在群益官網下載API套件壓縮檔裡。

[登入+報價/歷史K程式碼](D:\\Python\\Chen\\群益PythonAPI\\群益PythonAPI\\取報價或歷史K.py)

[登入+下單程式碼](D:\\Python\\Chen\\群益PythonAPI\\群益PythonAPI\\下單.py)

[登入+回報程式碼](D:\\Python\\Chen\\群益PythonAPI\\群益PythonAPI\\回報.py)

**環境設定:**

透過Python開發，使用群益API COM需要安裝下面幾項Python套件（非群益提供）：

以下為建議使用版本，可以個人需求更新。

1. comtypes(1.1.4)
2. pip(9.0.1)
3. pytz(2018.3)
4. pywin32(222)

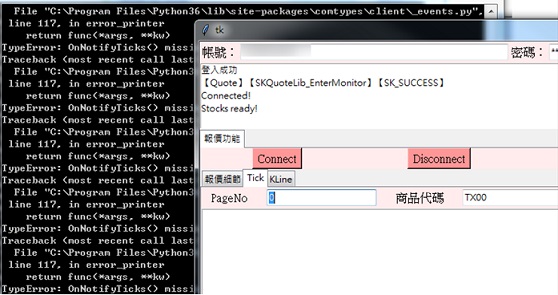
其中comtypes跟pywin32是一定需要的，另外使用的IDE是

Visual Studio Code。

SKAPI換版注意事項：

當使用comtypes時，換版後，部分功能應該會遇到以下問題，發生一直跳出錯誤訊息的情況，但程式卻又沒有錯誤！

※此例為OnNotifyTicks修改後發生錯誤



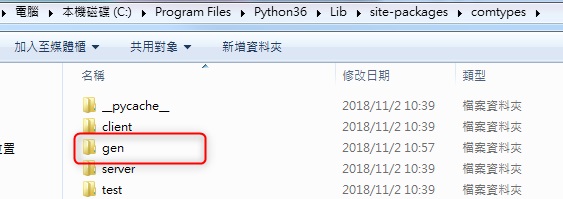
這是由於comtypes設計所致，當COM元件換版，若function或callback有修改的話，comtypes並不會同步更新相關檔案，python程式在執行的時候還是依據舊版本API的參數去處理，而無法正確對應API實際使用的參數，導致此問題發生！

因此我們需要將comtypes所產生的相關檔案作刪除動作，才能讓他重新產生新版COM元件所需的檔案。

1.從錯誤訊息中我們可以確認comtypes實際路徑如下



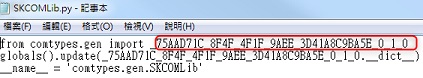
2.使用檔案總管至該目錄下可看到gen資料夾



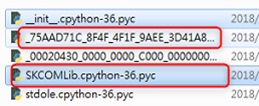
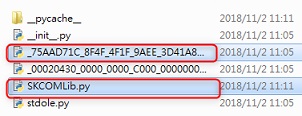
3.刪除相關檔案，下方兩種方式擇一

3-1.若comtypes相關程式只有SKAPI的話，可以直接砍掉整個gen資料夾！

3-2.若要個別刪除，不是只有SKCOMLib.py要刪除，需要先查看該檔內容



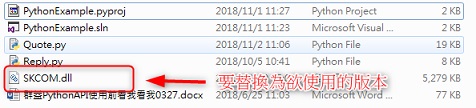
需要連這個GUID名稱的檔案也一起刪除掉，稍後重新產生才不會有問題



1. 產生相關檔案，請注意將此段code加入執行，這段code會產生comtypes相關檔案，產生後即可註解該行，因為產生檔案之後就不會再作用了。

comtypes.client.GetModule(os.path.split(os.path.realpath(\_\_file\_\_))[0] + r'\SKCOM.dll') #加此行需將API放與py同目錄

※注意python程式目錄也要放SKCOM.dll，這是要產生comtypes相關檔案所使用的。



1. 執行後，已可正常處理資料！

