策略王COM元件使用說明

-下單準備介紹-

## 函式範例

### SKOrderLib (下單)

*函式*

1. 下單物件初始化

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **初始** | [SKOrderLib\_Initialize](#_SKOrderLib_Initialize) | 需先執行才可執行相關下單函式。 |

1. 取得交易帳號

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **取得交易帳號** | [GetUserAccount](#_GetUserAccount) | * 下單帳號由該函式取得為主。 * 登入前，需先簽署證券或期貨API下單聲明書，方能取得相關市場帳號。 * 若未簽署，請簽署完成，再重新登入。 |

1. 讀取憑證資訊

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **讀取憑證** | [ReadCertByID](#_ReadCertByID) | 驗證憑證有效後，方可進行各相關委託功能。 |

1. Proxy連線/斷線/重新連線

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **初始** | [SKOrderLib\_InitialProxyByID](#_SKOrderLib_InitialProxyByID) | 需先執行才可執行相關Proxy下單函式。  (會先連線proxy server並做登入) |
| **斷線** | [ProxyDisconnectByID](#_ProxyDisconnectByID) | 主動斷線proxy server |
| **重新連線** | [ProxyReconnectByID](#_ProxyReconnectByID) | 重新連線proxy server |

1. 登入帳號查詢

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **查詢登入類型** | [SKOrderLib\_GetLoginType](#_SKOrderLib_GetLoginType) |  |
| **查詢下單類型** | [SKOrderLib\_GetSpeedyType](#_SKOrderLib_GetSpeedyType) |  |

1. 回報查詢

|  |  |  |
| --- | --- | --- |
| **功能** | **事件名稱** | **備註** |
| **委託回報查詢** | [GetOrderReport](#_GetOrderReport) |  |
| **成交回報查詢** | [GetFulfillReport](#_GetFulfillReport) |  |

1. 設定每秒下單限制

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **限制量設定** | [SetMaxQty](#_SetMaxQty) |  |
| **限制筆數設定** | [SetMaxCount](#_SetMaxCount) |  |
| **下單解鎖** | [UnlockOrder](#_UnlockOrder) |  |

1. 指令測試主機是否正常

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **Ping 和 Tracert主機測試** | [SKOrderLib\_PingandTracertTest](#_SKOrderLib_PingandTracertTest) | 適用於出現微軟socket error  確認與主機連線狀態 |
| **Telnet 主機測試** | [SKOrderLib\_TelnetTest](#_SKOrderLib_TelnetTest) | 測試主機連線是否正常 |

1. LOG上傳

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **上傳LOG** | [SKOrderLib\_LogUpload](#_SKOrderLib_LogUpload) | 上傳近3日LOG |

*事件*

|  |  |  |
| --- | --- | --- |
| **功能** | **事件名稱** | **備註** |
| 1. **帳號資料** | [OnAccount](#_OnAccount) |  |
| 1. **Proxy Server連線、登入狀態** | [OnProxyStatus](#_OnProxyStatus) |  |
| 1. **Telnet測試** | [OnTelnetTest](#_OnTelnetTest) |  |

##### 下單物件初始化

###### **SKOrderLib\_Initialize**

|  |  |  |
| --- | --- | --- |
| SKOrderLib 下單物件初始化。産生下單物件後需先執行初始動作。 | | |
| **宣告** | Long SKOrderLib\_Initialize(); | |
| **參數** | 無。 |  |
| **回傳值** | 0表示初始化成功，其餘非0數值都表示初始失敗。錯誤代碼可參考對照表。 | |
| **備註** |  | |

|  |
| --- |
| Python範例程式碼\_LoginForm.py 類別 OrderForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # buttonSKOrderLib\_Initialize          self.buttonSKOrderLib\_Initialize = tk.Button(self)          self.buttonSKOrderLib\_Initialize["text"] = "下單物件初始化"          self.buttonSKOrderLib\_Initialize["command"] = self.buttonSKOrderLib\_Initialize\_Click          self.buttonSKOrderLib\_Initialize.grid(column = 0, row = 1) |

|  |
| --- |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKOrder = comtypes.client.CreateObject(sk.SKOrderLib,interface=sk.ISKOrderLib) |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKOrderLib\_Initialize\_Click(self):          nCode = m\_pSKOrder.SKOrderLib\_Initialize()          msg = "【SKOrderLib\_Initialize】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

##### 取得交易帳號

###### **GetUserAccount**

|  |  |  |
| --- | --- | --- |
| 取回目前可交易的所有帳號。資料由[OnAccount](#_4-2-a_OnAccount)事件回傳。 | | |
| **宣告** | Long GetUserAccount(); | |
| **參數** | 無。 |  |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** | * 登入前，需先簽署證券或期貨API下單聲明書，方能取得相關市場帳號。 * 若未簽署，請先簽署完成，再重新登入。 | |

|  |
| --- |
| Python範例程式碼\_LoginForm.py 類別 OrderForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # buttonGetUserAccount          self.buttonGetUserAccount = tk.Button(self)          self.buttonGetUserAccount["text"] = "取回交易帳號"          self.buttonGetUserAccount["command"] = self.buttonGetUserAccount\_Click          self.buttonGetUserAccount.grid(column = 0, row = 2) |

|  |
| --- |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKOrder = comtypes.client.CreateObject(sk.SKOrderLib,interface=sk.ISKOrderLib) |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonGetUserAccount\_Click(self):          nCode = m\_pSKOrder.GetUserAccount()          msg = "【GetUserAccount】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

##### 讀取憑證資訊

###### **ReadCertByID**

|  |  |  |
| --- | --- | --- |
| 讀取憑證資訊。委託下單必須透過憑證，因此當元件初始化成功後即需要做讀取憑證的動作，如果使用群組的帳號做初始，則必須自行將所有的帳號依序做讀取憑證的動作。  如果送出委託前未經讀取憑證，送委託會得到 SK\_ERROR\_ORDER\_SIGN\_INVALID 的錯誤。 | | |
| **宣告** | Long ReadCertByID([in] BSTR bstrLogInID); | |
| **參數** | bstrLogInID | 登入帳號。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** |  | |

|  |
| --- |
| Python範例程式碼\_LoginForm.py 類別 OrderForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # buttonReadCertByID          self.buttonReadCertByID = tk.Button(self)          self.buttonReadCertByID["text"] = "讀取憑證"          self.buttonReadCertByID["command"] = self.buttonReadCertByID\_Click          self.buttonReadCertByID.grid(column = 0, row = 3) |

|  |
| --- |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKOrder = comtypes.client.CreateObject(sk.SKOrderLib,interface=sk.ISKOrderLib) |
| Current Block(放置於def createWidgets(self): 的同一層)，comboBoxUserID抓取使用者ID，詳細作法可參考範例程式 |
| def buttonReadCertByID\_Click(self):          nCode = m\_pSKOrder.ReadCertByID(comboBoxUserID.get())          msg = "【ReadCertByID】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

##### Proxy連線/斷線/重新連線

###### **SKOrderLib\_InitialProxyByID**

|  |  |  |
| --- | --- | --- |
| 以使用者的id，初使化proxy 下單連線 | | |
| **宣告** | Long SKOrderLib\_InitialProxyByID ([in] BSTR bstrLogInID) | |
| **參數** | bstrLogInID | 登入ID。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。 | |
| **備註** | * 若回傳值為0，需同時確認[OnProxyStatus](#_4-2-b_OnAsyncOrder) 收到5001，代表Proxy下單連線且登入成功 * 若回傳非0，請重新執行SKOrderLib\_InitialProxyByID。 | |

|  |
| --- |
| Python範例程式碼\_LoginForm.py 類別 OrderForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # buttonSKOrderLib\_InitialProxyByID          self.buttonSKOrderLib\_InitialProxyByID = tk.Button(self)          self.buttonSKOrderLib\_InitialProxyByID["text"] = "Proxy初始/連線"          self.buttonSKOrderLib\_InitialProxyByID["command"] = self.buttonSKOrderLib\_InitialProxyByID\_Click          self.buttonSKOrderLib\_InitialProxyByID.grid(column = 0, row = 4) |

|  |
| --- |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKOrder = comtypes.client.CreateObject(sk.SKOrderLib,interface=sk.ISKOrderLib) |
| Current Block(放置於def createWidgets(self): 的同一層)，comboBoxUserID抓取使用者ID，詳細作法可參考範例程式 |
| def buttonSKOrderLib\_InitialProxyByID\_Click(self):          nCode = m\_pSKOrder.SKOrderLib\_InitialProxyByID(comboBoxUserID.get())          msg = "【SKOrderLib\_InitialProxyByID】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

###### **ProxyDisconnectByID**

|  |  |  |
| --- | --- | --- |
| 以使用者的id，主動斷線proxy server的連線 | | |
| **宣告** | Long ProxyDisconnectByID ([in] BSTR bstrLogInID) | |
| **參數** | bstrLogInID | 登入ID。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。 | |
| **備註** |  | |

|  |
| --- |
| Python範例程式碼\_LoginForm.py 類別 OrderForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # buttonProxyDisconnectByID          self.buttonProxyDisconnectByID = tk.Button(self)          self.buttonProxyDisconnectByID["text"] = "Proxy斷線"          self.buttonProxyDisconnectByID["command"] = self.buttonProxyDisconnectByID\_Click          self.buttonProxyDisconnectByID.grid(column = 1, row = 4) |

|  |
| --- |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKOrder = comtypes.client.CreateObject(sk.SKOrderLib,interface=sk.ISKOrderLib) |
| Current Block(放置於def createWidgets(self): 的同一層)，comboBoxUserID抓取使用者ID，詳細作法可參考範例程式 |
| def buttonProxyDisconnectByID\_Click(self):          nCode = m\_pSKOrder.ProxyDisconnectByID(comboBoxUserID.get())          msg = "【ProxyDisconnectByID】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

###### **ProxyReconnectByID**

|  |  |  |
| --- | --- | --- |
| 以使用者的id，重新連線之前主動斷線的proxy server連線，並做login | | |
| **宣告** | Long ProxyReconnectByID ([in] BSTR bstrLogInID) | |
| **參數** | bstrLogInID | 登入ID。 |
| **回傳值** | 0表示連線成功，其餘非0數值都表示失敗。 | |
| **備註** |  | |

|  |
| --- |
| Python範例程式碼\_LoginForm.py 類別 OrderForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # buttonProxyReconnectByID          self.buttonProxyReconnectByID = tk.Button(self)          self.buttonProxyReconnectByID["text"] = "Proxy重新連線"          self.buttonProxyReconnectByID["command"] = self.buttonProxyReconnectByID\_Click          self.buttonProxyReconnectByID.grid(column = 2, row = 4) |

|  |
| --- |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKOrder = comtypes.client.CreateObject(sk.SKOrderLib,interface=sk.ISKOrderLib) |
| Current Block(放置於def createWidgets(self): 的同一層)，comboBoxUserID抓取使用者ID，詳細作法可參考範例程式 |
| def buttonProxyReconnectByID\_Click(self):          nCode = m\_pSKOrder.ProxyReconnectByID(comboBoxUserID.get())          msg = "【ProxyReconnectByID】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

##### 登入帳號查詢

###### **SKOrderLib\_GetLoginType**

|  |  |  |
| --- | --- | --- |
| 查詢登入帳號類型。 | | |
| **宣告** | Long SKOrderLib\_GetLoginType([in] BSTR bstrLogInID); | |
| **參數** | bstrLogInID | 登入帳號。 |
| **回傳值** | 0：一般帳號，1：VIP帳號。 | |
| **備註** |  | |

|  |
| --- |
| Python範例程式碼\_LoginForm.py 類別 OrderForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # buttonSKOrderLib\_GetLoginType          self.buttonSKOrderLib\_GetLoginType = tk.Button(self)          self.buttonSKOrderLib\_GetLoginType["text"] = "查詢登入帳號"          self.buttonSKOrderLib\_GetLoginType["command"] = self.buttonSKOrderLib\_GetLoginType\_Click          self.buttonSKOrderLib\_GetLoginType.grid(column = 0, row = 5) |

|  |
| --- |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKOrder = comtypes.client.CreateObject(sk.SKOrderLib,interface=sk.ISKOrderLib) |
| Current Block(放置於def createWidgets(self): 的同一層)，comboBoxUserID抓取使用者ID，詳細作法可參考範例程式 |
| def buttonSKOrderLib\_GetLoginType\_Click(self):          nCode = m\_pSKOrder.SKOrderLib\_GetLoginType(comboBoxUserID.get())          if nCode == 0:              msg = "一般帳號"          else:              msg = "VIP帳號"          msg = "【SKOrderLib\_GetLoginType】" + msg          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

###### **SKOrderLib\_GetSpeedyType**

|  |  |  |
| --- | --- | --- |
| 查詢登入帳號下單線路。 | | |
| **宣告** | Long SKOrderLib\_GetSpeedyType([in] BSTR bstrLogInID); | |
| **參數** | bstrLogInID | 登入帳號。 |
| **回傳值** | 0：一般線路，1：Speedy線路。 | |
| **備註** |  | |

|  |
| --- |
| Python範例程式碼\_LoginForm.py 類別 OrderForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # buttonSKOrderLib\_GetSpeedyType          self.buttonSKOrderLib\_GetSpeedyType = tk.Button(self)          self.buttonSKOrderLib\_GetSpeedyType["text"] = "查詢下單線路"          self.buttonSKOrderLib\_GetSpeedyType["command"] = self.buttonSKOrderLib\_GetSpeedyType\_Click          self.buttonSKOrderLib\_GetSpeedyType.grid(column = 0, row = 6) |

|  |
| --- |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKOrder = comtypes.client.CreateObject(sk.SKOrderLib,interface=sk.ISKOrderLib) |
| Current Block(放置於def createWidgets(self): 的同一層)，comboBoxUserID抓取使用者ID，詳細作法可參考範例程式 |
| def buttonSKOrderLib\_GetSpeedyType\_Click(self):          nCode = m\_pSKOrder.SKOrderLib\_GetSpeedyType(comboBoxUserID.get())          if nCode == 0:              msg = "一般線路"          else:              msg = "Speedy線路"          msg = "【SKOrderLib\_GetSpeedyType】" + msg          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

##### 回報查詢

###### **GetOrderReport**

|  |  |  |
| --- | --- | --- |
| 委託回報查詢 | | |
| **宣告** | void GetOrderReport([in] BSTR bstrLogInID, [in] BSTR strFullAccount, [in] LONG nFormat, [**out,retval**] BSTR\* bstrResult); | |
| **參數** | bstrLogInID | 登入帳號。 |
| bstrFullAccount | 完整帳號。  分公司代號 + 交易帳號 |
| nFormat | 查詢種類  1:全部 2:有效 3:可消 4:已消 5:已成 6:失敗7:合併同價格 8:合併同商品 9:預約 |
| bstrResult | 委託回報內容 |
| **回傳值** | 委託回報內容  回傳格式：以 ”,” 當分隔符號，以 ”0x0D 0x0A”(\r\n)當每筆的段落。  [回傳格式參考](#_證期委託查詢回報格式)  非正常委託內容時  M999: 查詢錯誤  M003: 查無資料 | |
| **備註** | 1. 查詢為阻塞式，如為避免查詢造成應用程式停頓，請以執行緒方式進行查詢 2. 限制每次查詢間需間隔五秒 3. 回報不含盤中零股 4. 回傳內容如有中文，編碼為 UTF-8 5. 市場別為國內證期選 | |

|  |
| --- |
| Python範例程式碼\_LoginForm.py 類別 OrderForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # comboBoxGetOrderReport          tk.Label(self, text = "查詢種類").grid(column = 0, row = 7)              #輸入框          self.comboBoxGetOrderReport = ttk.Combobox(self, state='readonly')          self.comboBoxGetOrderReport['values'] = Config.comboBoxGetOrderReport          self.comboBoxGetOrderReport.grid(column = 1, row = 7)          global comboBoxGetOrderReport          comboBoxGetOrderReport = self.comboBoxGetOrderReport              # buttonGetOrderReport          self.buttonGetOrderReport = tk.Button(self)          self.buttonGetOrderReport["text"] = "委託回報查詢"          self.buttonGetOrderReport["command"] = self.buttonGetOrderReport\_Click          self.buttonGetOrderReport.grid(column = 2, row = 7) |

|  |
| --- |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKOrder = comtypes.client.CreateObject(sk.SKOrderLib,interface=sk.ISKOrderLib) |
| Config.py |
| comboBoxGetOrderReport = (      "1:全部", "2:有效", "3:可消", "4:已消", "5:已成", "6:失敗", "7:合併同價格", "8:合併同商品", "9:預約"  ) |
| Current Block(放置於def createWidgets(self): 的同一層)，comboBoxUserID抓取使用者ID、 comboBoxAccount抓取交易帳號，詳細作法可參考範例程式 |
| def buttonGetOrderReport\_Click(self):          if comboBoxGetOrderReport.get() == "1:全部":              format = 1          elif comboBoxGetOrderReport.get() == "2:有效":              format = 2          elif comboBoxGetOrderReport.get() == "3:可消":              format = 3          elif comboBoxGetOrderReport.get() == "4:已消":              format = 4          elif comboBoxGetOrderReport.get() == "5:已成":              format = 5          elif comboBoxGetOrderReport.get() == "6:失敗":              format = 6          elif comboBoxGetOrderReport.get() == "7:合併同價格":              format = 7          elif comboBoxGetOrderReport.get() == "8:合併同商品":              format = 8          elif comboBoxGetOrderReport.get() == "9:預約":              format = 9          else:              format = 1          bstrResult = m\_pSKOrder.GetOrderReport(comboBoxUserID.get(), comboBoxAccount.get(), format)          msg = "【GetOrderReport】" + bstrResult          richTextBoxMessage.insert('end',  msg + "\n")          richTextBoxMessage.see('end') |

證期委託查詢回報格式

1. [Format] : [1]全部 [2] 有效 [3]可消 [4]已消 [5]已成 [6]失敗 [9]預約

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 屬性名稱 | 備註 | |
| 1 | 市場別 | TW:國內證期選 TS：國內證券 TF：國內期選 OS:複委託 OF:海期選 | |
| 2 | 商品別 | STO：股票 FUT：期貨 OPT：選擇權 ASO:資產交換選擇權 | |
| 3 | 交易所別 | TSEA：上市 TSEB：上櫃 OTC：興櫃 TAIFEX:期交所 HK:港股 HKEX港期 | |
| 4 | 分公司代號 | 例如 證券：9101 期選：F020000 | |
| 5 | IBNO | | |
| 6 | 交易帳號 | | |
| 7 | 子帳號 | | |
| 8 | 委託書號 | | |
| 9 | 13碼電子流水號 | | |
| 10 | 原始13碼電子流水號 | | |
| 11 | 委託狀態 | | 0：預約 2：全部成交 3：全部取消 4：部分成交，剩餘已取消 5：部分成交，剩餘可取消 6：委託失敗 7:委託成功 8:取消失敗 F:動態退單 9：取消中  【交易所動態退單為F開頭編碼】  F1：動態退單-全部取消 F2：動態退單-部分成交，剩餘已取消 F3：動態退單-部分委託成功 F4：否決 |
| 12 | 委託日期 | | 格式yyyyMMdd |
| 13 | 委託時間 | | 格式hhmmss，證券逐筆搓合為hhmmssfff |
| 14 | 交易歸屬日 | | 格式yyyyMMdd |
| 15 | 委託有效日 | | 格式yyyyMMdd |
| 16 | 商品代號 | | |
| 17 | Tandem商品代號1 | | |
| 18 | Tandem契約年月1 | | |
| 19 | Tandem履約價1 | | |
| 20 | Tandem商品代號2 | | |
| 21 | Tandem契約年月2 | | |
| 22 | Tandem履約價2 | | |
| 23 | 買賣別 | | B:買 S:賣/出借 |
| 24 | 盤別 | | A：一般 B：盤後 C：零股 D：拍賣 E：鉅額 F：盤中零股 G:標借  H:標購 I:證金標購 |
| 25 | 國內證券委託條件 | | 0:現股 1:代資 2:代券 3:融資 4:融券 5:一般策略性借券賣出 6:策略性交易借券賣出 8:無券賣出 |
| 26 | 委託條件 | | 0：ROD(DAY) (證券逐筆)1：GOOD\_TILL\_CANCEL 2：AT\_THE\_OPENING  3：IOC(證券逐筆) 4：FOK (證券逐筆) 5：GOOD\_TILL\_GROSSING 6：GOOD\_TILL\_DATE 7：AT\_THE\_CLOSE |
| 27 | 委託方式 | | 1：市價單(MKT) (證券逐筆) 2：限價單(LMT) (證券逐筆)  3：國內期貨-範圍市價 海外期貨-停損單(STP)4：停損限價單(STP LIMIT) 5：收市價(MOC)  P.S.策略王拿到3時會備註”範圍市價”。 |
| 28 | \*委託價 | | |
| 29 | 原始委託價格 | | |
| 30 | 有效委託數量 | | (1)原委託量-已改量的數量-刪單數量(2)動態退單：為餘量 |
| 31 | 原始委託數量 | | |
| 32 | 成交數量 | | |
| 33 | 剩餘數量 | | |
| 34 | 當沖註記 | | Y：當沖 N：新倉 O：平倉 A：自動 |
| 35 | 判別T+1盤 | | A:T盤 B:T+1盤 |
| 36 | 是否錯誤回報 | | Y:錯誤回報 N:非錯誤回報 |
| 37 | 下單來源別 | | |
| 38 | 交易單位股數 | | |
| 39 | 證券預約單價格註記 | | 證券只有預約單才有此欄位 空白：用委託價格 0：close price(前日收盤價) 1：bull price (漲停) 2：bear price (跌停) h：漲停下一檔 l(小寫L)：跌停上一檔 C：漲 1/2 c：跌 1/2 |
| 40 | 營業員代號 | | |
| 41 | Call/Put欄位 | | (僅選擇權) Call:C Put:P 非選擇權：空白 |
| 42 | 委託價-分子 | | (僅海期選) |
| 43 | 委託價-分母 | | (僅海期選) |
| 44 | 成交均價-價格 | | (僅海期選) 債券130'05(130) |
| 45 | 成交均價-分子 | | (僅海期選) 債券130'05(05) |
| 46 | 成交均價-分母 | | (僅海期選) 債券130'05(32) |
| 47 | 觸發價-價格 | | (僅海期選、複委託) |
| 48 | 觸發價-分子 | | (僅海期選) |
| 49 | 觸發價-分母 | | (僅海期選) |
| 50 | 交易所回覆訊息 | | 交易所回覆訊息 (回報格式 ExchMessage tag 1081 ) |
| 51 | 動態退單基準價 | | 退單基準價 |
| 52 | 取消總量 | | (僅證券)其它交易給0 |
| 53 | 成交日期 | | 格式yyyyMMdd |
| 54 | 成交時間 | | |
| 55 | 專戶註記 | | (僅複委託) |
| 56 | 多幣別註記 | | (僅複委託) |
| 57 | 交割幣別1 | | (僅複委託) |
| 58 | 交割幣別2 | | (僅複委託) |
| 59 | 交割幣別3 | | (僅複委託) |
| 60 | 6碼委託序號 | | (僅複委託) |
| 61 | 上手書號 | | 交易所原始委託序號DMA ORDER ID (目前適用SGX DMA) |
| 62 | 庫存別 | | (僅複委託) 庫存別 1：[美股]一般/定股(CITI)2：[美股]定額(VIEWTRADE) 3：[其它股市] 一般 4：[基金] 一般/定額 |
| 63 | 有效委託數量 | | (複委託碎股小數)  (1)原委託量-已改量的數量-刪單數量(2)動態退單：為餘量 |
| 64 | 原始委託數量 | | (複委託碎股小數) |
| 65 | 成交數量 | | (複委託碎股小數) |
| 66 | 剩餘數量 | | (複委託碎股小數) |
| 67 | 取消總量 | | (複委託碎股小數) |
| 68 | 委託時間 | | 格式HHmmssfff，證券秒下三位(毫秒)。 |
| 69 | 交易註記 | | 100000：智慧單  100001：智慧單拆單  200000：一般拆單 |
| 70 | 客製化商品註記 | | Y：是  空白：否 |

[Format]: [7]合併同價格

|  |  |  |
| --- | --- | --- |
| 編號 | 屬性名稱 | 備註 |
| 1 | 市場別 | TW:國內證期選 TS：國內證券 TF：國內期選 OS:複委託 OF:海期選 |
| 2 | 商品別 | STO：股票 FUT：期貨 OPT：選擇權 ASO:資產交換選擇權 |
| 3 | 交易所別 | TSEA：上市 TSEB：上櫃 OTC：興櫃 TAIFEX:期交所 |
| 4 | 分公司代號 | 例如 證券：9101 期選：F020000 |
| 5 | IBNO | |
| 6 | 交易帳號 | |
| 7 | 子帳號 | |
| 8 | 委託日期  交易歸屬日 | 格式yyyyMMdd  格式yyyyMMdd(交易歸屬日僅限複委託) |
| 9 | 商品代號 | |
| 11 | Tandem商品代號1 | |
| 12 | Tandem契約年月1 | |
| 13 | Tandem履約價1 | |
| 14 | Tandem商品代號2 | |
| 15 | Tandem契約年月2 | |
| 16 | Tandem履約價2 | |
| 17 | 買賣別 | B:買 S:賣/出借 |
| 18 | 盤別 | A：一般 B：盤後 C：零股 D：拍賣 E：鉅額 G:標借  H:標購 I:證金標購 |
| 19 | 國內證券委託條件 | 0:現股 1:代資 2:代券 3:融資 4:融券 5:一般策略性借券賣出 6:策略性交易借券賣出 8:無券賣出 |
| 20 | 委託條件 | 0：ROD(DAY) (證券逐筆)1：GOOD\_TILL\_CANCEL 2：AT\_THE\_OPENING  3：IOC(證券逐筆) 4：FOK (證券逐筆) 5：GOOD\_TILL\_GROSSING 6：GOOD\_TILL\_DATE 7：AT\_THE\_CLOSE P.S.策略王拿到3時會備註”範圍市價”。 |
| 21 | 委託價 | 僅[Format]: [7] 有 |
| 22 | 有效委託數量 | 原委託量扣除已改量的數量 |
| 23 | 原始委託數量 | |
| 24 | 成交量 | |
| 25 | 當沖註記 | Y：當沖 N：新倉 O：平倉 A：自動 |
| 26 | 營業員代號 | |
| 27 | 委託價-分子 | (僅海期選) |
| 28 | 委託價-分母 | (僅海期選) |
| 29 | 交易單位股數 | |
| 30 | 有效委託數量 | (複委託碎股小數)  原委託量扣除已改量的數量 |
| 31 | 原始委託數量 | (複委託碎股小數) |
| 32 | 成交數量 | (複委託碎股小數) |

1. [Format]: [8]合併同商品

|  |  |  |
| --- | --- | --- |
| 編號 | 屬性名稱 | 備註 |
| 1 | 市場別 | TW:國內證期選 TS：國內證券 TF：國內期選 OS:複委託 OF:海期選 |
| 2 | 商品別 | STO：股票 FUT：期貨 OPT：選擇權 ASO:資產交換選擇權 |
| 3 | 交易所別 | TSEA：上市 TSEB：上櫃 OTC：興櫃 TAIFEX:期交所 |
| 4 | 分公司代號 | 例如 證券：9101 期選：F020000 |
| 5 | IBNO | |
| 6 | 交易帳號 | |
| 7 | 子帳號 | |
| 8 | 委託日期  交易歸屬日 | 格式yyyyMMdd  格式yyyyMMdd(交易歸屬日僅限複委託) |
| 9 | 商品代號 | |
| 10 | 商品名稱 | |
| 11 | Tandem商品代號1 | |
| 12 | Tandem契約年月1 | |
| 13 | Tandem履約價1 | |
| 14 | Tandem商品代號2 | |
| 15 | Tandem契約年月2 | |
| 16 | Tandem履約價2 | |
| 17 | 買賣別 | B:買 S:賣/出借 |
| 18 | 盤別 | A：一般 B：盤後 C：零股 D：拍賣 E：鉅額 G:標借  H:標購 I:證金標購 |
| 19 | 國內證券委託條件 | 0:現股 1:代資 2:代券 3:融資 4:融券 5:一般策略性借券賣出 6:策略性交易借券賣出 8:無券賣出 |
| 20 | 委託條件 | 0：ROD(DAY) (證券逐筆)1：GOOD\_TILL\_CANCEL 2：AT\_THE\_OPENING  3：IOC(證券逐筆) 4：FOK (證券逐筆) 5：GOOD\_TILL\_GROSSING 6：GOOD\_TILL\_DATE 7：AT\_THE\_CLOSE P.S.策略王拿到3時會備註”範圍市價”。 |
| 21 | 委託價 | 僅[Format]: [7] 有 |
| 22 | 有效委託數量 | 原委託量扣除已改量的數量 |
| 23 | 原始委託數量 | |
| 24 | 成交量 | |
| 25 | 當沖註記 | Y：當沖 N：新倉 O：平倉 A：自動 |
| 26 | 營業員代號 | |
| 27 | 交易單位股數 | |
| 28 | 有效委託數量 | (複委託碎股小數)  原委託量扣除已改量的數量 |
| 29 | 原始委託數量 | (複委託碎股小數) |
| 30 | 成交量 | (複委託碎股小數) |

###### **GetFulfillReport**

|  |  |  |
| --- | --- | --- |
| 成交回報查詢 | | |
| **宣告** | void GetFulfillReport([in] BSTR bstrLogInID, [in] BSTR strFullAccount, [in] LONG nFormat, [**out,retval**] BSTR\* bstrResult); | |
| **參數** | bstrLogInID | 登入帳號。 |
| bstrFullAccount | 完整帳號。  分公司代號 + 交易帳號 |
| nFormat | 查詢種類  1:完整 2:合併同書號 3:合併同價格 4:合併同商品  5:T+1成交回報 |
| bstrResult | 委託回報內容 |
| **回傳值** | 成交回報內容  回傳格式：以 ”,” 當分隔符號，以 ”0x0D 0x0A”(\r\n)當每筆的段落。  [回傳格式參考](#_證期成交查詢回報格式)  非正常委託內容時  M999: 查詢錯誤  M003: 查無資料 | |
| **備註** | 1. 查詢為阻塞式，如為避免查詢造成應用程式停頓，請以執行緒方式進行查詢 2. 限制每次查詢間需間隔五秒 3. 回報不含盤中零股 4. 回傳內容如有中文，編碼為 UTF-8 5. 市場別為國內證期選 | |

|  |
| --- |
| Python範例程式碼\_LoginForm.py 類別 OrderForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # comboBoxGetFulfillReport          tk.Label(self, text = "查詢種類").grid(column = 0, row = 8)              #輸入框          self.comboBoxGetFulfillReport = ttk.Combobox(self, state='readonly')          self.comboBoxGetFulfillReport['values'] = Config.comboBoxGetFulfillReport          self.comboBoxGetFulfillReport.grid(column = 1, row = 8)          global comboBoxGetFulfillReport          comboBoxGetFulfillReport = self.comboBoxGetFulfillReport              # buttonGetFulfillReport          self.buttonGetFulfillReport = tk.Button(self)          self.buttonGetFulfillReport["text"] = "成交回報查詢"          self.buttonGetFulfillReport["command"] = self.buttonGetFulfillReport\_Click          self.buttonGetFulfillReport.grid(column = 2, row = 8) |

|  |
| --- |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKOrder = comtypes.client.CreateObject(sk.SKOrderLib,interface=sk.ISKOrderLib) |
| Config.py |
| comboBoxGetFulfillReport = (      "1:全部", "2:合併同書號", "3:合併同價格", "4:合併同商品", "5:T+1成交回報"  ) |
| Current Block(放置於def createWidgets(self): 的同一層)，comboBoxUserID抓取使用者ID、 comboBoxAccount抓取交易帳號，詳細作法可參考範例程式 |
| def buttonGetFulfillReport\_Click(self):          if comboBoxGetFulfillReport.get() == "1:全部":              format = 1          elif comboBoxGetFulfillReport.get() == "2:合併同書號":              format = 2          elif comboBoxGetFulfillReport.get() == "3:合併同價格":              format = 3          elif comboBoxGetFulfillReport.get() == "4:合併同商品":              format = 4          elif comboBoxGetFulfillReport.get() == "5:T+1成交回報":              format = 5          else:              format = 1          bstrResult = m\_pSKOrder.GetFulfillReport(comboBoxUserID.get(), comboBoxAccount.get(), format)          msg = "【GetFulfillReport】" + bstrResult          richTextBoxMessage.insert('end',  msg + "\n")          richTextBoxMessage.see('end') |

證期成交查詢回報格式

1. [Format] : [1]完整 [5]T+1成交回報

|  |  |  |
| --- | --- | --- |
| 編號 | 屬性名稱 | 備註 |
| 1 | 市場別 | TW:國內證期選 TS：國內證券 TF：國內期選 OS:複委託 OF:海期選 |
| 2 | 商品別 | STO：股票 FUT：期貨 OPT：選擇權 ASO:資產交換選擇權 |
| 3 | 交易所別 | TSEA：上市 TSEB：上櫃 OTC：興櫃 TAIFEX:期交所 |
| 4 | 分公司代號 | 例如 證券：9101 期選：F020000 |
| 5 | IBNO | |
| 6 | 交易帳號 | |
| 7 | 子帳號 | |
| 8 | 委託書號 | |
| 9 | 成交序號 | |
| 10 | 成交日期 | 格式yyyyMMdd |
| 11 | 成交時間 | 格式hhmmss，證券逐筆搓合為hhmmssfff |
| 12 | 交易歸屬日 | 格式yyyyMMdd |
| 13 | 商品代號 | |
| 14 | 商品名稱 | |
| 15 | Tandem商品代號1 | |
| 16 | Tandem契約年月1 | |
| 17 | Tandem履約價1 | |
| 18 | 成交價1 | |
| 19 | Tandem商品代號2 | |
| 20 | Tandem契約年月2 | |
| 21 | Tandem履約價2 | |
| 22 | 成交價2 | |
| 23 | 買賣別 | B:買 S:賣/出借 |
| 24 | 盤別 | A：一般 B：盤後 C：零股 D：拍賣 E：鉅額 G:標借  H:標購 I:證金標購 |
| 25 | 國內證券委託條件 | 0:現股 1:代資 2:代券 3:融資 4:融券 5:一般策略性借券賣出 6:策略性交易借券賣出 8:無券賣出 |
| 26 | 委託條件 | 0：ROD(DAY) (證券逐筆)1：GOOD\_TILL\_CANCEL 2：AT\_THE\_OPENING  3：IOC(證券逐筆) 4：FOK (證券逐筆) 5：GOOD\_TILL\_GROSSING 6：GOOD\_TILL\_DATE 7：AT\_THE\_CLOSE |
| 27 | 成交價 | |
| 28 | 成交量 | |
| 29 | 當沖註記 | Y：當沖 N：新倉 O：平倉 A：自動 |
| 30 | 判別T+1盤 | A:T盤 B:T+1盤 |
| 31 | 下單來源別 | |
| 32 | 委託方式 | |
| 33 | 委託有效日 | |
| 34 | 營業員代號 | |
| 35 | 手續費 | (僅證券、複委託)  證券：預估手續費 千分之1.425  複委託：TANDEM提供 |
| 36 | 交易稅 | (僅證券)預估交易稅 千分之1 或千分之3 |
| 37 | 幣別 | 提供給複委託使用的幣別，其餘皆代入空白。 |
| 38 | 委託日期 | |
| 39 | 委託時間 | |
| 40 | 成交價-分子 | |
| 41 | 成交價-分母 | |
| 42 | 交易單位股數 | |
| 43 | 專戶註記 | (僅複委託) |
| 44 | 多幣別註記 | (僅複委託) |
| 45 | 總成交價金 | |
| 46 | 6碼委託書號 | (僅複委託) |
| 47 | 上手書號 | 交易所原始委託序號DMA ORDER ID (目前適用SGX DMA) |
| 48 | 庫存別 | 庫存別 1：[美股]一般/定股(CITI)2：[美股]定額(VIEWTRADE) 3：[其它股市] 一般 4：[基金] 一般/定額 |
| 49 | 成交量 | (複委託碎股小數) |
| 50 | 前手息 | (複委託海外債前手息) |
| 51 | 成交時間 | 格式HHmmssfff，證券秒下三位(毫秒)。 |
| 52 | 客製化商品註記 | Y:是  空白:否 |

[Format]:[2]合併同書號

|  |  |  |
| --- | --- | --- |
| 編號 | 屬性名稱 | 備註 |
| 1 | 市場別 | TW:國內證期選 TS：國內證券 TF：國內期選 OS:複委託 OF:海期選 |
| 2 | 商品別 | STO：股票 FUT：期貨 OPT：選擇權 ASO:資產交換選擇權 |
| 3 | 交易所別 | TSEA：上市 TSEB：上櫃 OTC：興櫃 TAIFEX:期交所 |
| 4 | 分公司代號 | 例如 證券：9101 期選：F020000 |
| 5 | IBNO | |
| 6 | 交易帳號 | |
| 7 | 子帳號 | |
| 8 | 委託書號 | |
| 9 | 交易歸屬日 | 格式yyyyMMdd |
| 10 | 商品代號 | |
| 11 | 商品名稱 | |
| 12 | Tandem商品代號1 | |
| 13 | Tandem契約年月1 | |
| 14 | Tandem履約價1 | |
| 15 | 成交價1 | |
| 16 | Tandem商品代號2 | |
| 17 | Tandem契約年月2 | |
| 18 | Tandem履約價2 | |
| 19 | 成交價2 | |
| 20 | 買賣別 | B:買 S:賣/出借 |
| 21 | 盤別 | A：一般 B：盤後 C：零股 D：拍賣 E：鉅額 G:標借  H:標購 I:證金標購 |
| 22 | 國內證券委託條件 | 0:現股 1:代資 2:代券 3:融資 4:融券 5:一般策略性借券賣出 6:策略性交易借券賣出 8:無券賣出 |
| 23 | 委託條件 | 0：ROD(DAY) (證券逐筆)1：GOOD\_TILL\_CANCEL 2：AT\_THE\_OPENING  3：IOC(證券逐筆) 4：FOK (證券逐筆) 5：GOOD\_TILL\_GROSSING 6：GOOD\_TILL\_DATE 7：AT\_THE\_CLOSE |
| 24 | 成交價 | |
| 25 | 成交量 | |
| 26 | 當沖註記 | Y：當沖 N：新倉 O：平倉 A：自動 |
| 27 | 判別T+1盤 | A:T盤 B:T+1盤 |
| 28 | 委託有效日 | |
| 29 | 營業員代號 | |
| 30 | 幣別 | 提供給複委託使用的幣別，其餘皆代入空白。 |
| 31 | 成交價-分子 | |
| 32 | 成交價-分母 | |
| 33 | 交易單位股數 | |
| 34 | 手續費 | (僅證券、複委託)  證券：預估手續費 千分之1.425  複委託：TANDEM提供 |
| 35 | 總成交價金 | (僅複委託) |
| 36 | 六碼委託書號 | (僅複委託) |
| 37 | 成交量 | (複委託碎股小數) |

[Format]:[3]合併同價格 [4]合併同商品

|  |  |  |
| --- | --- | --- |
| 編號 | 屬性名稱 | 備註 |
| 1 | 市場別 | TW:國內證期選 TS：國內證券 TF：國內期選 OS:複委託 OF:海期選 |
| 2 | 商品別 | STO：股票 FUT：期貨 OPT：選擇權 ASO:資產交換選擇權 |
| 3 | 交易所別 | TSEA：上市 TSEB：上櫃 OTC：興櫃 TAIFEX:期交所 |
| 4 | 分公司代號 | 例如 證券：9101 期選：F020000 |
| 5 | IBNO | |
| 6 | 交易帳號 | |
| 7 | 子帳號 | |
| 8 | 交易歸屬日 | 格式yyyyMMdd |
| 9 | 商品代號 | |
| 10 | 商品名稱 | |
| 11 | Tandem商品代號1 | |
| 12 | Tandem契約年月1 | |
| 13 | Tandem履約價1 | |
| 14 | 成交價1 | 僅[Format]: [3] 有 |
| 15 | Tandem商品代號2 | |
| 16 | Tandem契約年月2 | |
| 17 | Tandem履約價2 | |
| 18 | 成交價2 | 僅[Format]: [3] 有 |
| 19 | 買賣別 | B:買 S:賣/出借 |
| 20 | 盤別 | A：一般 B：盤後 C：零股 D：拍賣 E：鉅額 G:標借  H:標購 I:證金標購 |
| 21 | 國內證券委託條件 | 0:現股 1:代資 2:代券 3:融資 4:融券 5:一般策略性借券賣出 6:策略性交易借券賣出 8:無券賣出 |
| 22 | 委託條件 | 0：ROD(DAY) (證券逐筆)1：GOOD\_TILL\_CANCEL 2：AT\_THE\_OPENING  3：IOC(證券逐筆) 4：FOK (證券逐筆) 5：GOOD\_TILL\_GROSSING 6：GOOD\_TILL\_DATE 7：AT\_THE\_CLOSE |
| 23 | 成交價 | [Format]: [4] 呈現平均價 |
| 24 | 成交量 | |
| 25 | 當沖註記 | Y：當沖 N：新倉 O：平倉 A：自動 |
| 26 | 判別T+1盤 | A:T盤 B:T+1盤 |
| 27 | 委託有效日 | |
| 28 | 營業員代號 | |
| 29 | 幣別 | 提供給複委託使用的幣別，其餘皆代入空白。 |
| 30 | 成交價-分子 | [Format]: [4] 呈現平均價 |
| 31 | 成交價-分母 | |
| 32 | 交易單位股數 | |
| 33 | 預估手續費 | (僅複委託) |
| 34 | 成交總價金 | (僅複委託) |
| 35 | 專戶註記 | (僅複委託) |
| 36 | 成交量 | (複委託碎股小數) |

一張含有 文字, 字型, 白色, 印刷術 的圖片

自動產生的描述

Data source: <https://www.capital.com.tw/event/stock/oddLot/default.asp>

##### 設定每秒下單限制

###### **SetMaxQty**

|  |  |  |
| --- | --- | --- |
| 設定**每秒**委託「量」限制。一秒內下單超過設定值時下該類型下單將被鎖定，需進行解鎖才可繼續下單。 | | |
| **宣告** | Long SetMaxQty( [in] LONG nMarketType, [in] LONG nMaxQty ); | |
| **參數** | nMarketType | 0：TS(證券)  1：TF(期貨)  2：TO(選擇權)  3：OS(複委託)  4：OF(海外期貨)  5：OO(海外選擇權) |
| nMaxQty | 委託量，小於等於零無限制。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** | 若下單超過設定被上鎖時，需呼叫 [UnlockOrder](#_UnlockOrder) 解鎖，才可繼續下單。 | |

|  |
| --- |
| Python範例程式碼\_LoginForm.py 類別 OrderForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # comboBoxSetMaxQtynMarketType          tk.Label(self, text="市場種類").grid(column=0, row=9)              # 輸入框          self.comboBoxSetMaxQtynMarketType = ttk.Combobox(self, state='readonly')          self.comboBoxSetMaxQtynMarketType['values'] = Config.comboBoxSetMaxQtynMarketType          self.comboBoxSetMaxQtynMarketType.grid(column=1, row=9)          global comboBoxSetMaxQtynMarketType          comboBoxSetMaxQtynMarketType = self.comboBoxSetMaxQtynMarketType              # Entry          tk.Label(self, text="委託量").grid(column=2, row=9)          self.entrySetMaxQtynMaxQty = tk.Entry(self)          self.entrySetMaxQtynMaxQty.grid(column=3, row=9)          global entrySetMaxQtynMaxQty          entrySetMaxQtynMaxQty = self.entrySetMaxQtynMaxQty              # buttonSetMaxQty          self.buttonSetMaxQty = tk.Button(self)          self.buttonSetMaxQty["text"] = "每秒委託「量」限制"          self.buttonSetMaxQty["command"] = self.buttonSetMaxQty\_Click          self.buttonSetMaxQty.grid(column=4, row=9) |

|  |
| --- |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKOrder = comtypes.client.CreateObject(sk.SKOrderLib,interface=sk.ISKOrderLib) |
| Config.py |
| comboBoxSetMaxQtynMarketType =(       "0：TS(證券)", "1：TF(期貨)", "2：TO(選擇權)", "3：OS(複委託)", "4：OF(海外期貨)", "5：OO(海外選擇權)"  ) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSetMaxQty\_Click(self):          if comboBoxSetMaxQtynMarketType.get() == "0：TS(證券)":              nMarketType = 0          elif comboBoxSetMaxQtynMarketType.get() == "1：TF(期貨)":              nMarketType = 1          elif comboBoxSetMaxQtynMarketType.get() == "2：TO(選擇權)":              nMarketType = 2          elif comboBoxSetMaxQtynMarketType.get() == "3：OS(複委託)":              nMarketType = 3          elif comboBoxSetMaxQtynMarketType.get() == "4：OF(海外期貨)":              nMarketType = 4          elif comboBoxSetMaxQtynMarketType.get() == "5：OO(海外選擇權)":              nMarketType = 5          else:              nMarketType = 0          nCode = m\_pSKOrder.SetMaxQty(nMarketType, int(entrySetMaxQtynMaxQty.get()))          msg = "【SetMaxQty】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

###### **SetMaxCount**

|  |  |  |
| --- | --- | --- |
| 設定**每秒**委託「筆數」限制。一秒內下單超過設定值時下該類型下單將被鎖定，需進行解鎖才可繼續下單。 | | |
| **宣告** | Long SetMaxCount( [in] LONG nMarketType, [in] LONG nMaxCount ); | |
| **參數** | nMarketType | 0：TS (證券)  1：TF(期貨)  2：TO(選擇權)  3：OS(複委託)  4：OF(海外期貨)  5：OO(海外選擇權) |
| nMaxCount | 委託筆數，小於等於零無限制。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** | 若下單超過設定被上鎖時，需呼叫 [UnlockOrder](#_UnlockOrder) 解鎖，才可繼續下單。 | |

|  |
| --- |
| Python範例程式碼\_LoginForm.py 類別 OrderForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # comboBoxSetMaxCountnMarketType          tk.Label(self, text="市場種類").grid(column=0, row=10)              # 輸入框          self.comboBoxSetMaxCountnMarketType = ttk.Combobox(self, state='readonly')          self.comboBoxSetMaxCountnMarketType['values'] = Config.comboBoxSetMaxCountnMarketType          self.comboBoxSetMaxCountnMarketType.grid(column=1, row=10)          global comboBoxSetMaxCountnMarketType          comboBoxSetMaxCountnMarketType = self.comboBoxSetMaxCountnMarketType              # Entry          tk.Label(self, text="委託筆數").grid(column=2, row=10)          self.entrySetMaxCountnMaxCount = tk.Entry(self)          self.entrySetMaxCountnMaxCount.grid(column=3, row=10)          global entrySetMaxCountnMaxCount          entrySetMaxCountnMaxCount = self.entrySetMaxCountnMaxCount              # buttonSetMaxCount          self.buttonSetMaxCount = tk.Button(self)          self.buttonSetMaxCount["text"] = "每秒委託「筆數」限制"          self.buttonSetMaxCount["command"] = self.buttonSetMaxCount\_Click          self.buttonSetMaxCount.grid(column=4, row=10) |

|  |
| --- |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKOrder = comtypes.client.CreateObject(sk.SKOrderLib,interface=sk.ISKOrderLib) |
| Config.py |
| comboBoxSetMaxCountnMarketType =(       "0：TS(證券)", "1：TF(期貨)", "2：TO(選擇權)", "3：OS(複委託)", "4：OF(海外期貨)", "5：OO(海外選擇權)"  ) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSetMaxCount\_Click(self):          if comboBoxSetMaxCountnMarketType.get() == "0：TS(證券)":              nMarketType = 0          elif comboBoxSetMaxCountnMarketType.get() == "1：TF(期貨)":              nMarketType = 1          elif comboBoxSetMaxCountnMarketType.get() == "2：TO(選擇權)":              nMarketType = 2          elif comboBoxSetMaxCountnMarketType.get() == "3：OS(複委託)":              nMarketType = 3          elif comboBoxSetMaxCountnMarketType.get() == "4：OF(海外期貨)":              nMarketType = 4          elif comboBoxSetMaxCountnMarketType.get() == "5：OO(海外選擇權)":              nMarketType = 5          else:              nMarketType = 0          nCode = m\_pSKOrder.SetMaxCount(nMarketType, int(entrySetMaxCountnMaxCount.get()))          msg = "【SetMaxCount】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

###### **UnlockOrder**

|  |  |  |
| --- | --- | --- |
| 下單解鎖。下單函式上鎖後需經由此函式解鎖才可繼續下單。 | | |
| **宣告** | Long UnlockOrder([in] LONG nMarketType); | |
| **參數** | nMarketType | 0：TS (證券)  1：TF(期貨)  2：TO(選擇權)  3：OS(複委託)  4：OF(海外期貨)  5：OO(海外選擇權) |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。 | |
| **備註** |  | |

|  |
| --- |
| Python範例程式碼\_LoginForm.py 類別 OrderForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # comboBoxUnlockOrder          tk.Label(self, text = "市場種類").grid(column = 0, row = 11)              #輸入框          self.comboBoxUnlockOrder = ttk.Combobox(self, state='readonly')          self.comboBoxUnlockOrder['values'] = Config.comboBoxUnlockOrder          self.comboBoxUnlockOrder.grid(column = 1, row = 11)          global comboBoxUnlockOrder          comboBoxUnlockOrder = self.comboBoxUnlockOrder              # buttonUnlockOrder          self.buttonUnlockOrder = tk.Button(self)          self.buttonUnlockOrder["text"] = "下單解鎖"          self.buttonUnlockOrder["command"] = self.buttonUnlockOrder\_Click          self.buttonUnlockOrder.grid(column = 2, row = 11) |

|  |
| --- |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKOrder = comtypes.client.CreateObject(sk.SKOrderLib,interface=sk.ISKOrderLib) |
| Config.py |
| comboBoxUnlockOrder =(       "0：TS(證券)", "1：TF(期貨)", "2：TO(選擇權)", "3：OS(複委託)", "4：OF(海外期貨)", "5：OO(海外選擇權)"  ) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonUnlockOrder\_Click(self):          if comboBoxUnlockOrder.get() == "0：TS(證券)":              nMarketType = 0          elif comboBoxUnlockOrder.get() == "1：TF(期貨)":              nMarketType = 1          elif comboBoxUnlockOrder.get() == "2：TO(選擇權)":              nMarketType = 2          elif comboBoxUnlockOrder.get() == "3：OS(複委託)":              nMarketType = 3          elif comboBoxUnlockOrder.get() == "4：OF(海外期貨)":              nMarketType = 4          elif comboBoxUnlockOrder.get() == "5：OO(海外選擇權)":              nMarketType = 5          else:              nMarketType = 0          nCode = m\_pSKOrder.UnlockOrder(nMarketType)          msg = "【UnlockOrder】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

##### 指令測試主機是否正常

###### **SKOrderLib\_PingandTracertTest**

|  |  |  |
| --- | --- | --- |
| 測試API使用到的主機是否正常 | | |
| **宣告** | Long SKOrderLib\_PingandTracertTest(); | |
| **參數** |  |  |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。 | |
| **備註** | 適用於出現**微軟socket error**可以確認與主機連線狀態  可在LOG檔**「PingTest」**看到測試結果  請注意：檔案內看到TracertTest -> XXX 才算測試完成。  一張含有 文字, 字型, 數字, 軟體 的圖片  自動產生的描述 | |

|  |
| --- |
| Python範例程式碼\_LoginForm.py 類別 OrderForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # buttonSKOrderLib\_PingandTracertTest          self.buttonSKOrderLib\_PingandTracertTest = tk.Button(self)          self.buttonSKOrderLib\_PingandTracertTest["text"] = "Ping測試API使用到的主機是否正常"          self.buttonSKOrderLib\_PingandTracertTest["command"] = self.buttonSKOrderLib\_PingandTracertTest\_Click          self.buttonSKOrderLib\_PingandTracertTest.grid(column = 0, row = 12) |

|  |
| --- |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKOrder = comtypes.client.CreateObject(sk.SKOrderLib,interface=sk.ISKOrderLib) |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKOrderLib\_PingandTracertTest\_Click(self):          nCode = m\_pSKOrder.SKOrderLib\_PingandTracertTest()          msg = "【SKOrderLib\_PingandTracertTest】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

###### **SKOrderLib\_TelnetTest**

|  |  |  |
| --- | --- | --- |
| 用Telnet指令測試與API使用到的主機連線是否正常。 | | |
| **宣告** | Long SKOrderLib\_TelnetTest (); | |
| **參數** |  |  |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** | 結果由 [OnTelnetTest](#_OnTelnetTest) 事件回傳。 | |

|  |
| --- |
| Python範例程式碼\_LoginForm.py 類別 OrderForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # buttonSKOrderLib\_TelnetTest          self.buttonSKOrderLib\_TelnetTest = tk.Button(self)          self.buttonSKOrderLib\_TelnetTest["text"] = "Telnet測試API使用到的主機是否正常"          self.buttonSKOrderLib\_TelnetTest["command"] = self.buttonSKOrderLib\_TelnetTest\_Click          self.buttonSKOrderLib\_TelnetTest.grid(column = 0, row = 13) |

|  |
| --- |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKOrder = comtypes.client.CreateObject(sk.SKOrderLib,interface=sk.ISKOrderLib) |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKOrderLib\_TelnetTest\_Click(self):          nCode = m\_pSKOrder.SKOrderLib\_TelnetTest()          msg = "【SKOrderLib\_TelnetTest】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

##### LOG上傳

###### **SKOrderLib\_LogUpload**

|  |  |  |
| --- | --- | --- |
| LOG上傳。 | | |
| **宣告** | Long SKOrderLib\_LogUpload(); | |
| **參數** |  |  |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** | 可用此功能上傳近3日LOG。  上傳後，LOG路徑會生成一個資料夾和一個zip，資料夾內為上傳的檔案。  一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片  自動產生的描述 | |

|  |
| --- |
| Python範例程式碼\_LoginForm.py 類別 OrderForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # buttonSKOrderLib\_LogUpload          self.buttonSKOrderLib\_LogUpload = tk.Button(self)          self.buttonSKOrderLib\_LogUpload["text"] = "打包近3日LOG"          self.buttonSKOrderLib\_LogUpload["command"] = self.buttonSKOrderLib\_LogUpload\_Click          self.buttonSKOrderLib\_LogUpload.grid(column = 0, row = 14) |

|  |
| --- |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKOrder = comtypes.client.CreateObject(sk.SKOrderLib,interface=sk.ISKOrderLib) |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKOrderLib\_LogUpload\_Click(self):          nCode = m\_pSKOrder.SKOrderLib\_LogUpload()          msg = "【SKOrderLib\_LogUpload】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

## 事件

### OnAccount

|  |  |  |
| --- | --- | --- |
| 帳號資訊。透過呼叫[GetUserAccount](#_GetUserAccount)後，帳號資訊由該事件回傳。 | | |
| **宣告** | void OnAccount([in] BSTR bstrLogInID, [in] BSTR bstrAccountData); | |
| **參數** | bstrLogInID | 登入ID |
| bstrAccountData | 帳號資訊。以逗點分隔每一個欄位，欄位依序為：  『市場,分公司代碼,分公司,帳號,身份證字號,姓名』  \*備註  證券：分公司代碼；期貨：Broker id |
| **備註** | * 登入前，需先簽署證券或期貨API下單聲明書，方能取得相關市場帳號。 * 若未簽署，請先簽署完成，再重新登入。 | |

|  |
| --- |
| Python範例程式碼\_LoginForm.py |
| UI程式碼(放置於 def createWidgets(self): 的下一層)/UI示意圖 |
| # richTextBoxMessage          self.richTextBoxMessage = tk.Listbox(self, height=5)          self.richTextBoxMessage.grid(column = 0, row = 10, columnspan=5, sticky = tk.E + tk.W)          global richTextBoxMessage          richTextBoxMessage = self.richTextBoxMessage |

|  |
| --- |
| 事件物件註冊(紅色底部份，黑色底為該事件，這裡用OnAccount示範)，此為 SKOrderLibEvent，依照不同物件名做修改 |
| # OrderLib事件  class SKOrderLibEvent():      # 帳號資訊      def OnAccount(self, bstrLogInID, bstrAccountData):          msg = "【OnAccount】" + bstrLogInID + "\_" + bstrAccountData;          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end')  SKOrderEvent = SKOrderLibEvent();  SKOrderLibEventHandler = comtypes.client.GetEvents(m\_pSKOrder, SKOrderEvent); |
| 事件(可放在事件註冊物件底下一層) |
| # 帳號資訊      def OnAccount(self, bstrLogInID, bstrAccountData):          msg = "【OnAccount】" + bstrLogInID + "\_" + bstrAccountData;          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

### OnProxyStatus

|  |  |  |
| --- | --- | --- |
| 一個使用者id會與proxy server建一條連線，此事件回傳此條連線的連線狀態。  透過呼叫 [**SKOrderLib\_InitialProxyByID**](#_SKOrderLib_InitialProxyByID) 後，資訊由該事件回傳。 | | |
| **宣告** | void OnProxyStatus ([in] BSTR bstrUserId, [in]LONG nCode); | |
| **參數** | bstrUserId | 回傳內容：使用者id |
| nCode | 回傳內容：  請參考代碼定義表5001~5006、5011 |
| **備註** | 一張含有 文字, 字型, 螢幕擷取畫面, 收據 的圖片  自動產生的描述 | |

|  |
| --- |
| Python範例程式碼\_LoginForm.py |
| UI程式碼(放置於 def createWidgets(self): 的下一層)/UI示意圖 |
| # richTextBoxMessage          self.richTextBoxMessage = tk.Listbox(self, height=5)          self.richTextBoxMessage.grid(column = 0, row = 10, columnspan=5, sticky = tk.E + tk.W)          global richTextBoxMessage          richTextBoxMessage = self.richTextBoxMessage |

|  |
| --- |
| 事件物件註冊(紅色底部份，黑色底為該事件，這裡用OnAccount示範)，此為 SKOrderLibEvent，依照不同物件名做修改 |
| # OrderLib事件  class SKOrderLibEvent():      # 帳號資訊      def OnAccount(self, bstrLogInID, bstrAccountData):          msg = "【OnAccount】" + bstrLogInID + "\_" + bstrAccountData;          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end')  SKOrderEvent = SKOrderLibEvent();  SKOrderLibEventHandler = comtypes.client.GetEvents(m\_pSKOrder, SKOrderEvent); |
| 事件(可放在事件註冊物件底下一層) |
| # 一個使用者id會與proxy server建一條連線，此事件回傳此條連線的連線狀態      def OnProxyStatus(self, bstrUserId, nCode):          msg = "【OnProxyStatus】" + bstrUserId + "\_" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode);          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

### OnTelnetTest

|  |  |  |
| --- | --- | --- |
| 透過呼叫 [SKOrderLib\_TelnetTest](#_SKOrderLib_TelnetTest) 後，資訊由該事件回傳。 | | |
| **宣告** | void OnTelnetTest ([in] BSTR bstrData); | |
| **參數** | bstrData | 回傳內容：   * 測試中 * 共測試x台主機 : 成功y台, 失敗z台 |
| **備註** |  | |

|  |
| --- |
| Python範例程式碼\_LoginForm.py |
| UI程式碼(放置於 def createWidgets(self): 的下一層)/UI示意圖 |
| # richTextBoxMessage          self.richTextBoxMessage = tk.Listbox(self, height=5)          self.richTextBoxMessage.grid(column = 0, row = 10, columnspan=5, sticky = tk.E + tk.W)          global richTextBoxMessage          richTextBoxMessage = self.richTextBoxMessage |

|  |
| --- |
| 事件物件註冊(紅色底部份，黑色底為該事件，這裡用OnAccount示範)，此為 SKOrderLibEvent，依照不同物件名做修改 |
| # OrderLib事件  class SKOrderLibEvent():      # 帳號資訊      def OnAccount(self, bstrLogInID, bstrAccountData):          msg = "【OnAccount】" + bstrLogInID + "\_" + bstrAccountData;          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end')  SKOrderEvent = SKOrderLibEvent();  SKOrderLibEventHandler = comtypes.client.GetEvents(m\_pSKOrder, SKOrderEvent); |
| 事件(可放在事件註冊物件底下一層) |
| # 透過呼叫 SKOrderLib\_TelnetTest 後，資訊由該事件回傳      def OnTelnetTest(self, bstrData):          msg = "【OnTelnetTest】" + bstrData;          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

## 錯誤代碼定義表

|  |  |  |
| --- | --- | --- |
| **代號** | **名稱** | **說明** |
| 1000 | SK\_ERROR\_LOGIN\_FIRST | 請先由Center 進行登入動作  \*請注意登入帳號是否為**大寫** |
| 1001 | SK\_ERROR\_INITIALIZE\_FAIL | 登入失敗，請由LOG查看失敗原因。 |
| 1002 | SK\_ERROR\_ACCOUNT\_NOT\_EXIST | 交易帳號不存在。 |
| 1003 | SK\_ERROR\_ACCOUNT\_MARKET\_NOT\_MATCH | 下單帳號類型不正確，是否有證券帳號下期貨之情行。 |
| 1004 | SK\_ERROR\_PERIOD\_OUT\_OF\_RANGE | PERIOD 超出選擇範圍。 |
| 1005 | SK\_ERROR\_FLAG\_OUT\_OF\_RANGE | FLAG 超出選擇範圍。 |
| 1006 | SK\_ERROR\_BUYSELL\_OUT\_OF\_RANGE | BUYSELL 超出選擇範圍。 |
| 1007 | SK\_ERROR\_ORDER\_SERVER\_INVALID | 下單主機不存在。 |
| 1009 | SK\_ERROR\_TRADE\_TYPR\_OUT\_OF\_RANGE | TRADE TYPE 超出選擇範圍。 |
| 1010 | SK\_ERROR\_DAY\_TRADE\_OUT\_OF\_RANGE | DAY TRADE超出選擇範圍。 |
| 1011 | SK\_ERROR\_ORDER\_SIGN\_INVALID |  |
| 1012 | SK\_ERROR\_NEW\_CLOSE\_OUT\_OF\_RANGE | NEW CLOSE 超出選擇範圍。 |
| 1013 | SK\_ERROR\_PRODUCT\_INVALID | 下單商品不存在。 |
| 1014 | SK\_ERROR\_QTY\_INVALID | 單量錯誤，請確認是否為數字。 |
| 1015 | SK\_ERROR\_DAYTRADE\_DENIED | 商品不支援當沖 |
| 1016 | SK\_ERROR\_SPCIAL\_TRADE\_TYPE\_INVALID | 商品不支援該下單類型。 |
| 1017 | SK\_ERROR\_PRICE\_INVALID | 下單價格錯誤。 |
| 1018 | SK\_ERROR\_INDEX\_OUT\_OF\_RANGE | INDEX 超出選擇範圍。 |
| 1019 | SK\_ERROR\_QUERY\_IN\_PROCESSING | 查詢尚未結束請稍後再試。 |
| 1022 | SK\_ERROR\_FUNCTION\_PERMISSION\_DENIED |  |
| 1023 | SK\_ERROR\_MARKET\_OUT\_OF\_RANGE | MARKET 超出選擇範圍。 |
| 1025 | SK\_ERROR\_FOREIGNSTOCK\_PRICE\_OUT\_OF\_RANGE |  |
| 1026 | SK\_ERROR\_FOREIGNSTOCK\_UNDEFINE\_COINTYPE |  |
| 1027 | SK\_ERROR\_FOREIGNSTOCK\_SAME\_COINSTYPE |  |
| 1028 | SK\_ERROR\_FOREIGNSTOCK\_SALE\_SHOULD\_ORIGINAL\_COIN |  |
| 1029 | SK\_ERROR\_FOREIGNSTOCK\_TRADE\_UNIT\_INVALID |  |
| 1030 | SK\_ERROR\_FOREIGNSTOCK\_STOCKNO\_INVALID |  |
| 1031 | SK\_ERROR\_FOREIGNSTOCK\_ACCOUNTTYPE\_INVALID |  |
| 1032 | SK\_ERROR\_FOREIGNSTOCK\_INITIALIZE\_FAIL |  |
| 1033 | SK\_ERROR\_TS\_INITIALIZE\_FAIL |  |
| 1034 | SK\_ERROR\_OVERSEA\_TRADE\_PRODUCT\_FAIL |  |
| 1035 | SK\_ERROR\_OVERSEA\_TRADE\_DATA\_NOT\_COMPLETE | 海期交易商品尚未初使完成，請先執行SKOrderLib\_LoadOSCommodity |
| 1036 | SK\_ERROR\_CERT\_VERIFY\_CN\_INVALID | 憑證CN錯誤， |
| 1037 | SK\_ERROR\_CERT\_VERIFY\_SERVER\_REJECT | 憑證驗章失敗，請確認憑證有效性 |
| 1038 | SK\_ERROR\_CERT\_NOT\_VERIFIED | 憑證尚未驗章，請先執行ReadCertByID |
| 1039 | SK\_ERROR\_OVERSEA\_ACCOUNT\_NOT\_EXIST | 海期帳號不存在。 |
| 1040 | SK\_ERROR\_ORDER\_LOCK | 下單上鎖，請先執行UnlockOrder進行解鎖。 |
| 1041 | SK\_ERROR\_SERVER\_NOT\_CONNECTED | 與主機尚未連線。 |
| 1042 | SK\_ERROR\_OVERSEA\_KLINE\_DATA\_TYPE\_NOT\_FOUND | KLINE TYPE 超出選擇範圍。 |
| 1043 | SK\_ERROR\_STRIKEPRICE\_INVALID | 履約價不正確。 |
| 1044 | SK\_ERROR\_CALLPUT\_INVALID | CALL PUT 不正確。 |
| 1045 | SK\_ERROR\_CERT\_NOT\_FOUND | 憑證不存在，請先匯入憑證到IE 中。 |
| 1046 | SK\_ERROR\_RESERVED\_OUT\_OF\_RANGE | 委託盤別超出範圍 |
| 1047 | SK\_ERROR\_SMART\_TRADE\_ORDER\_SERVER\_INVALID | 智慧單中台主機不存在 |
| 1048 | SK\_ERROR\_CANCEL\_ORDER\_SMARTKEY\_INVALID | 智慧單(停損單)序號為無效值 |
| 1049 | SK\_ERROR\_OVERSEA\_FUTURE\_SPREAD\_ORDER | 海期價差委託時，（價格為非０數值）出現分子填入負號之錯誤 |
| 1050 | SK\_ERROR\_MIT\_ORDER\_EXCLUDE\_SPREAD\_PRODUCT | 智慧單MIT不可委託價差商品 |
| 1051 | SK\_ERROR\_SGX\_API\_ORDER\_SEQNO\_CONFILICT | SGX API 委託序號重覆 |
| 1052 | SK\_ERROR\_OVERSEA\_FUTURE\_SPREAD\_ORDER\_NO\_DAYTRADE | 海期價差委託不可帶當沖 |
| 1053 | SK\_ERROR\_SGX\_API\_LOGON\_FAIL | SGX API登入失敗 |
| 1054 | SK\_ERROR\_MIT\_ORDER\_MUST\_GIVE\_TRIGGER\_PRICE | MIT委託內容需含觸發價 |
| 1056 | SK\_ERROR\_STRATEGY\_ORDER\_MUST\_GIVE\_DEAL\_PRICE | 智慧單委託內容需含成交價 |
| 1057 | SK\_ERROR\_WITHDRAW\_WITHOUT\_PASSWORD | 未輸入出入金密碼 |
| 1058 | SK\_ERROR\_WITHDRAWINOUT\_TYPE\_OUT\_OF\_RANGE | 出入金互轉類別錯誤 |
| 1059 | SK\_ERROR\_CURRENCY\_OUT\_OF\_RANGE | 幣別錯誤 |
| 1060 | SK\_ERROR\_AP\_GW\_SERVER\_INVALID | AP GW主機位址錯誤 |
| 1061 | SK\_ERROR\_STRATEGY\_ORDER\_BUYSELL\_INVALID | 證券智慧單買賣別錯誤 |
| 1062 | SK\_ERROR\_STRATEGY\_ORDER\_SMARTKEY\_TYPE\_INVALID | 證券智慧單號錯誤 |
| 1063 | SK\_ERROR\_STRATEGY\_ORDER\_MARKET\_TYPE\_INVALID | 證券智慧單市場代碼錯誤 |
| 1064 | SK\_ERROR\_STRATEGY\_ORDER\_TRADE\_TYPE\_INVALID | 證券智慧單單別錯誤 |
| 1065 | SK\_ERROR\_CORRECT\_PRICE\_ONLY\_LMT\_RANGE | 海期改價僅限限價單委託 |
| 1066 | SK\_ERROR\_TF\_OFFSET\_COMMODITY\_OUT\_OF\_RANGE | 期貨互抵商品別無效  (限定為0或1) |
| 1067 | SK\_ERROR\_SPECIAL\_TRADE\_TYPE\_OUT\_OF\_RANGE | (逐筆交易)委託價格類型有誤 |
| 1068 | SK\_ERROR\_SPECIAL\_TRADE\_TYPE\_IS\_MARKETPRICE\_AND\_ORDERPRICE\_SHOULD\_BE\_ZERO | (逐筆交易)凡市價單之委託價應為0 |
| 1069 | SK\_ERROR\_MARKET\_TYPE\_INVALID | 市場別有誤 |
| 1071 | SK\_ERROR\_CORRECT\_PRICE\_ONLY\_LMT\_RANGE | 海期改價單委託類型限LMT |
| 1072 | SK\_ERROR\_SGX\_DMA\_IS\_NOT\_ALLOW\_CORRECTPRICE\_BYBOOKNO | SGX專線無書號改價 |
| 1073 | SK\_ERROR\_SGX\_API\_ORDER\_SEQNO\_ERROR | SGX專線改價-委託序號錯誤 |
| 1076 | SK\_ERROR\_PROFITLOSSGWQUERY\_DATE\_INVALID | 損益查詢日期有誤 |
| 1077 | SK\_ERROR\_PROFITLOSSGWQUERY \_BOOKNO\_IS\_INVALID | 損益查詢書號有誤 |
| 1078 | SK\_ERROR\_PROFITLOSSGWQUERY\_SEQNO\_IS\_INVALID | 損益查詢序號有誤 |
| 1082 | SK\_ERROR\_THERE\_IS\_NO\_OVERSEA\_FUTURE\_ACCOUNT | 登入ID查無海期帳號 |
| 1083 | SK\_ERROR\_MOVE\_POINT\_IS\_**INVALID** | 移動點數有誤 |
| 1084 | SK\_ERROR\_TRADE\_KIND\_IS\_INVALID | 無效委託類別 |
| 1085 | SK\_ERROR\_PRICE\_TYPE\_IS\_INVALID | 無效價格型別 |
| 1086 | SK\_ERROR\_STRATEGY\_OUTORDER\_PRICE\_TYPE\_INVALID | 無效的出場單委託價格類別 |
| 1087 | SK\_ERROR\_DATE\_TYPE\_IS\_INVALID | 無效的日期格式 |
| 1088 | SK\_ERROR\_STRATEGY\_ORDER\_PRIME\_TYPE\_INVALID | 無效的股票市場類別 |
| 1089 | SK\_ERROR\_TRIGGER\_PRICE\_IS\_EQUAL\_TO\_DEAL | 指定觸發價等於成交價格無法觸發,請改觸發條件 |
| 1090 | SK\_ERROR\_SEQNO\_IS\_INVALID | 委託序號有誤 |
| 1091 | SK\_ERROR\_BOOKNO\_IS\_INVALID | 委託書號有誤 |
| 1092 | SK\_ERROR\_ORDER\_TYPE\_IS\_INVALID | 委託別有誤(EX:現股等) |
| 1093 | SK\_ERROR\_TRIGGER\_DIRECTION\_IS\_INVALID | 指定觸價方向有誤 |
| 1094 | SK\_ERROR\_TRIGGER\_BASE\_IS\_INVALID | 無效觸價基準；或未給觸價基準 |
| 1096 | SK\_ERROR\_TRIGGER\_METHOD\_IS\_INVALID | 選擇之觸發方式有誤 |
| 1100 | SK\_ERROR\_STRATEGYORDER\_OUT\_CONDICTION\_INVALID | 智慧單:未選擇至少一個出場條件 |
| 1101 | SK\_ERROR\_CANCEL\_ORDER\_FIELD\_INVALID | 智慧單刪單欄位值無效 |
| 1102 | SK\_ERROR\_ORDER\_FIELD\_INVALID | 智慧單出場條件或設定值無效 |
| 1104 | SK\_ERROR\_CANNOT\_GET\_DID\_FROM\_REGISTRY |  |
| 1105 | SK\_ERROR\_PRODUCT\_INVALID\_IN\_SGXTABLE | （SGX DMA專線用）  因專線商品清單查無對應商品，請確認該商品代碼是否有效或**洽交易室確認是否可網路交易** |
| 1106 | SK\_ERROR\_GET\_TANDEMINFO\_FAIL\_SRVKLINE | 無法轉後台商品代碼等資料，  可先確認委託商品代碼是否正確 |
| 1107 | SK\_ERROR\_LIMIT\_NEAR\_BY\_YEARMONTH | 限制近月商品代碼  EX: TX00、 MTX00 |
| 1110 | SK\_ERROR\_FOREIGNSTOCK\_ORDERTYPE\_INVALID | 複委託下單或刪單時，請填入委託類別：買、賣或刪單 |
| 1111 | SK\_ERROR\_FOREIGNSTOCK\_TRADETYPE\_INVALID | 複委託下單，請填入下單庫存類別 |
| 1112 | SK\_ERROR\_QUERY\_FORMAT\_INVALID | 需指定查詢格式  EX:新國內未平倉功能GW需指定格式 |
| 1121 | SK\_ERROR\_STRATEGY\_ORDER\_PRERISK\_INVALID | 無效預風控設定 |
| 1125 | SK\_ERROR\_TRIGGER\_CONDITION\_IS\_INVALID | 無效觸價條件；或未給觸價條件 |
| 1126 | SK\_ERROR\_EXCHANGE\_ID\_IS\_INVALID | 查無商品交易所代碼 |
| 1127 | SK\_ERROR\_QUERY\_IS\_OVER\_TEN\_TIMES | 一個account在1分鐘內不可超過10次 |
| 2001 | SK\_WARNING\_OF\_COM\_DATA\_MISSING | 海期交易商品檔下載失敗 |
| 2006 | SK\_WARNING\_ORDER\_DID\_NOT\_LOCKED | 下單市埸未上鎖。 |
| 2008 | SK\_WARNING\_OO\_COM\_ORDERDATA\_MISSING | 海選交易商品檔下載失敗 |
| 2009 | SK\_WARNING\_SIGN\_TS\_SMARTTRADE\_RISK\_NOITICE\_FIRST | 提醒簽署證券智慧單風險預告書 |
| 2010 | SK\_WARNING\_SIGN\_TF\_SMARTTRADE\_RISK\_NOITICE\_FIRST | 提醒簽署期貨智慧單風險預告書 |
| 2012 | SK\_WARNING\_OSQUOTECENTER\_IS\_NOT\_EXIST | 下單:下載海期/海選元件不存在 |
| 2013 | SK\_WARNING\_INITIALIZE\_OSQUOTECENTER\_CONNECTION\_FAIL | 下單:下載海期商品檔元件連線失敗 |
| 2014 | SK\_WARNING\_INITIALIZE\_OSQUOTECENTER\_OO\_CONNECTION\_FAIL | 下單:下載海選商品檔元件連線失敗 |
| 2015 | SK\_WARNING\_DOWNLOAD\_OF\_COM\_DATA\_IS\_TIMEOUT | 下單:下載海期商品檔未完成 |
| 2016 | SK\_WARNING\_DOWNLOAD\_OO\_COM\_DATA\_IS\_TIMEOUT | 下單:下載海選商品檔未完成 |
| 2030 | SK\_WARNING\_SHOULD\_FILLED\_YEARMONTH\_WHEN\_NOT\_NEAR\_MONTH | (非近月)  指定月份商品需填入商品契約年月共6碼 |
| 157 | SK\_ERROR\_LOGIN\_NO\_ACCONTS | 僅適用V2.13.45(含)以上版本  未能取得交易帳號 |
| 9998 | SK\_THIS\_FUNCTION\_NOT\_SUPPORTED | 此函式目前不提供 |

### Proxy Server 下單錯誤代碼

|  |  |  |
| --- | --- | --- |
| 5001 | SK\_PROXY\_SERVER\_LOGIN\_SUCCESS | Proxy handler連線中且狀態為登入成功 |
| 5002 | SK\_PROXY\_SERVER\_LOGIN\_FAIL | Proxy handler連線中且狀態為登入失敗  請重新連線ProxyReconnectByID |
| 5003 | SK\_PROXY\_SERVER\_DISCONNECT | Proxy handler斷線中  請重新連線  ProxyReconnectByID |
| 5004 | SK\_PROXY\_SERVER\_SCHEDULE\_DIALY\_DISCONNECT | Proxy SERVER送出每日斷線通知  請等待1分鐘後重新連線  ProxyReconnectByID |
| 5005 | SK\_PROXY\_SERVER\_SWITCH\_MODE | Proxy SERVER送出切轉主機通知 |
| 5006 | SK\_PROXY\_SERVER\_HANDLER\_IS\_EXIST | SKOrderLib\_InitialProxyByID時發生該ID連線HANDLER已存在  請重新連線  ProxyReconnectByID |
| 5009 | SK\_PROXY\_SERVER\_LOGIN\_SEND\_DATA\_SUCCESS | Proxy handler已連線且登入，等待Server回傳通知5001  ＊請確認收到5001再送proxy下單 |
| 5010 | SK\_PROXY\_SERVER\_CONNECT\_SUCCESS | Proxy handler連線中尚未登入  需斷線ProxyDisconnectByID成功，再重新連線ProxyReconnectByID |
| 5011 | SK\_PROXY\_SERVER\_CONNECT\_FAIL | Proxy handler連線失敗  請重新連線  ProxyReconnectByID  或初使化  SKOrderLib\_InitialProxyByID |
| 5012 | SK\_PROXY\_SERVER\_DISCONNECT\_FAIL | Proxy handler斷線失敗 |
| 5013 | SK\_PROXY\_LOAD\_PROXYLIB\_FAIL | 載入ProxyLIB DLL失敗  請先確認Proxy初使化  SKOrderLib\_InitialProxyByID |
| 5014 | SK\_PROXY\_SERVER\_CONNECT\_FAIL\_WITHOUT\_HANDLER | Proxy handler連線失敗  請重新連線  ProxyReconnectByID |
| 5015 | SK\_PROXY\_INITIALIZE\_FAIL | Proxy 初使化失敗  請重新連線  ProxyReconnectByID  或初使化  SKOrderLib\_InitialProxyByID |
| 5016 | SK\_PROXY\_ERROR\_THE\_ID\_IS\_CONNECTED\_ALREADY | 重新連線時發生重覆連線通知。  目前Proxy handler連線中且狀態為登入成功。 |
| 5017 | SK\_PROXY\_ERROR\_THE\_ID\_IS\_DISCONNECTED\_ALREADY | 斷線時，發生重覆斷線通知 |
| 5018 | SK\_PROXY\_SEND\_ORDER\_FAIL | 送出ProxyOrder異常  請確認SKOrderLib event : OnProxyStatus 為SK\_PROXY\_SERVER\_LOGIN\_SUCCESS  並重新連線  ProxyReconnectByID |
| 5019 | SK\_PROXY\_SERVER\_ACCOUNT\_RESTRICTED | 此帳號已被限制連線，請洽詢營業員 |