

## **A Study on Comparison of Open Application Programming Interface of Securities Companies Supporting Python**

Gui Yeol Ryu

*Professor, Department of Software, Seokyeong University, Seoul 02713, Korea*  
*e-mail: gyryu@skuniv.ac.kr*

### ***Abstract***

*Securities and investment services had the most data per company on the average, and used the most data. Investors are increasingly demanding to invest through their own analysis methods. Therefore, securities and investment companies provide stock data to investors through open API. The data received using the open API is in text format. Python is effective and convenient for requesting and receiving text data.*

*We investigate there are 22 major securities and investment companies in Korea and only 6 companies. Only Daishin Securities Co. supports Python officially. We compare how to receive stock data through open API using Python, and Python programming features. The open APIs for the study are Daishin Securities Co. and eBest Investment & Securities Co. Comparing the two APIs for receiving the current stock data, we find the main two differences are the login method and the method of sending and receiving data. As for the login method, CYBOS plus has login information, but xingAPI does not have. As for the method of sending and receiving data, Cybos Plus sends and receives data by calling the request method, and the reply method. xingAPI sends and receives data in the form of an event. Therefore, the number of xingAPI codes is more than that of CYBOS plus. And we find that CYBOS plus executes a loop statement by lists and tuple, dictionary, and CYBOS plus supports the basic commands provided by Python.*

**Keywords:** *Big Data, Open API, Python, Securities and investment services, Stock data*

### **1. Introduction**

The KB management Institute reported average data per company per industry in the US based on IDC and McKinsey & Company data, and securities and investment services had the most data per company on the average [1]. The McKinsey Global Institute categorized data produced and stored by industry sector and showed that securities and investment services used the most data [2]. They say systems for large data should be introduced in the securities and investment fields. Until now, the data of securities and investment services are extracted by storing and searching in DBMS. Securities and investment companies support home trading system (HTS) using DBMS. The data in the DBMS is suitable for searching and extracting, but not for analysis. And investors are increasingly demanding to invest through their own analysis methods [3]. Therefore, securities and investment companies provide stock-related data to investors through open Application Programming Interface (API). These were described as Fintech [4]. Big data is applied in various fields including machine learning and artificial intelligence [5, 6].

The data received using the open API is in text format. Python is effective and convenient for requesting

and receiving text data. This paper will investigate the current status of Korean securities companies' open API, and compare how to receive stock data through open API using Python and Python programming features. The open APIs for the study are Daishin Securities Co. and eBest Investment & Securities Co., because Daishin Securities Co. is the only open API that officially supports Python, and eBest Investment & Securities Co. can represent open API that unofficially supports Python.

As shown in Table 1, there are 22 major securities and investment companies in Korea such as SK Securities Co., Hanwha Investment & Securities Co., Shinyoung Securities Co., Bookook Securities Co., Hanyang Securities Co., Samsung Securities, Kyobo Securities Co., KB Securities Co., Hyundai Motor Securities, Daishin Securities Co., Hi Investment & Securities Co., NH Investment & Securities Co., Kiwoom Securities Co., eBest Investment & Securities Co., Mirae Asset Daewoo Co., Yuanta Securities Co., Eugene Investment Co., Korea Investment & Securities Co., KTB Investment & Securities Co., Hana Financial Investment Co., IBK Securities Co., and Cape Investment & Securities Co.. They support Home Trading System (HTS) which allows individuals to trade stocks anytime, anywhere by online. But only 6 companies support open API for big data. Table 1 shows the HTS and open API support status of 22 securities and investment companies in Korea [7].

**Table 1. open API supported by 22 major securities and investment companies in Korea**

Securities and investment services	HTS	Open API
SK Securities Co.	Yes	No
Hanwha Investment & Securities Co.	Yes	No
Shinyoung Securities Co.	Yes	No
Bookook Securities Co.	Yes	No
Hanyang Securities Co.	Yes	No
Samsung Securities	Yes	No
Kyobo Securities Co.	Yes	No
KB Securities Co.	Yes	No
Hyundai Motor Securities	Yes	No
Daishin Securities Co.	Yes	Yes
Hi Investment & Securities Co.	Yes	No
NH Investment & Securities Co.	Yes	No
Kiwoom Securities Co.	Yes	Yes
eBest Investment & Securities Co.	Yes	Yes
Mirae Asset Daewoo Co.	Yes	No
Yuanta Securities Co.	Yes	Yes
Eugene Investment Co.	Yes	No
Korea Investment & Securities Co.	Yes	No
KTB Investment & Securities Co.	Yes	No
Hana Financial Investment Co.	Yes	Yes
IBK Securities Co.	Yes	Yes
Cape Investment & Securities Co.	Yes	No
Total	22	6

Python is an important language for analysis of stock data, because it is a suitable language with various application packages for data science, machine learning applications, large-scale data processing, especially predictive analytics and more. Python is more convenient to use than other languages such as C, C++, and Java etc. The number of Python developers is increasing rapidly. Therefore, it is important that securities companies support Python.

The results of investigation on software supported by open API are as follows. Open API of Daishin Securities Co. supports Microsoft Visual Studio (VB, C#, C++, .Net etc.), and Microsoft Office (Word, Excel, Powerpoint, Access etc.), Microsoft Internet Explorer Dynamic HTML, Python, and Borland Delphi [8]. Open API of eBest Investment & Securities Co. supports VB, C/C++, C#, Delphi etc. [9]. Open API of Kiwoom Securities Co. supports VB, and C/C++, C#, Excel, Delphi etc. [10]. Open API of Yuanta Securities

Co. supports VB, and C/C++, C#, Delphi etc. [11]. Open API of Hana Financial Investment Co. supports MFC, and C#, VB, Delphi, Excel etc.[12]. Open API of IBK Securities Co. supports Microsoft Visual Studio (C++, VB, J++, etc.), and Microsoft Office (Word, Excel, Powerpoint, Access etc.), Microsoft Internet Explorer Dynamic HTML, Borland Delphi [13]. Only open API of Daishin Securities Co. supports Python officially. All open APIs support VB, C#, and Delphi. 5 open APIs except Hana Financial Investment Co. support C/C++. 4 open APIs except eBest Investment & Securities Co., and Yuanta Securities Co. support excel for ease of use. Only Daishin Securities Co. supports Python officially. But there was no open API supporting Java, which maybe becomes an obstacle to big data. Table 2 summarizes software supported by securities companies' open APIs.

**Table 2. Software supported by open API**

Securities and investment services	Supported Software	Supporting Python officially
Daishin Securities Co.	Microsoft Visual Studio (VB, C#, C++, .Net etc.), Microsoft Office (Word, Excel, Powerpoint, Access etc.), Microsoft Internet Explorer Dynamic HTML, Python, and Borland Delphi	Yes
Kiwoom Securities Co.	VB, C/C++, C#, Excel, and Delphi	No
eBest Investment & Securities Co.	VB, C/C++, C#, and Delphi	No
Yuanta Securities Co.	VB, C/C++, C#, and Delphi	No
Hana Financial Investment Co.	MFC, C#, VB, Delphi, Excel	No
IBK Securities Co.	Microsoft Visual Studio (C++, VB, J++, etc.), Microsoft Office (Word, Excel, Powerpoint, Access etc.), Microsoft Internet Explorer Dynamic HTML, Borland Delphi	No

## 2. Comparison of Two Open APIs using Python

### 2.1 How to Get Current Price by open API of Daishin Securities Co.

We can find how to access open API of Daishin Securities Co. on the website. To access the server of Daishin Securities Co., CYBOS plus which is the name of open API must be run in the background. We code in Python to access CYBOS plus, and CYBOS plus accesses Daishin Securities Co.'s server to receive data. CYBOS plus sends the result to Python. Since CYBOS plus checks the login permission, the login information is not required in the Python code.

Figure 1 shows the program for receiving a current price of the code "A000020". The first line imports the win32com.client module. The win32com is the library that helps to call ActiveX functions using Python on Windows. To create COM objects written in other programming languages in Python, we use the Dispatch method of a module called win32com.client. The second line creates an instance of the StockMst class by using the Dispatch method of the win32com.client module. The StockMst class is a class that receives the current price. The server name to connect is dscbo1. The third line gives the code value. The fourth line specifies the code property of the instance of the StockMst class. The parameters of the method SetInputValue mean that the value of code is designated as "A000020". The fifth line requests data processing. The sixth line prints the received current price along with the code value. For more details, please refer to the manual.

```

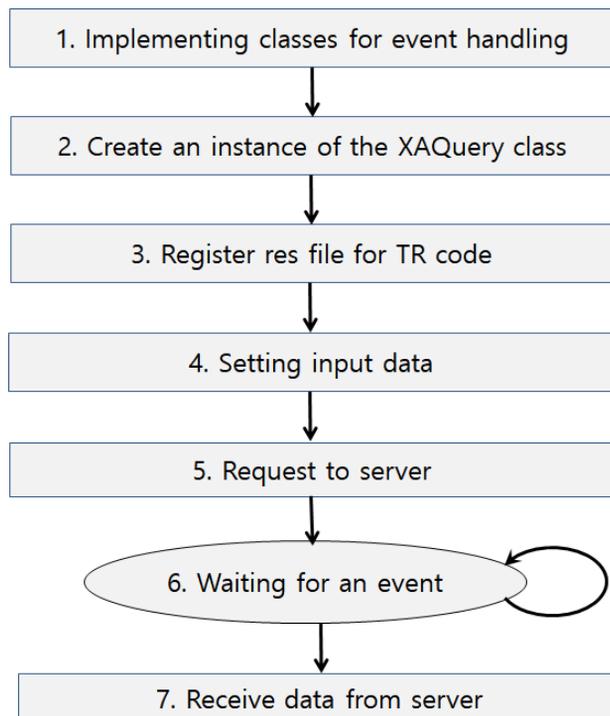
import win32com.client
instStockMst= win32com.client.Dispatch("dscbo1.StockMst")
code = "A000020"
instStockMst.SetInputValue(0, code)
instStockMst.BlockRequest()
print(code, instStockMst.GetHeaderValue(13))

```

**Figure 1. Python program of Daishin Securities Co. for current price**

## 2.2 How to Get Current Price by Open API of eBest Investment & Securities Co.

eBest Investment & Securities Co. provides xingAPI package for open API. xingAPI is the name of the open API of eBest Investment & Securities Co. xingAPI is a connection program that enables the trading program written by customers to send and receive data or trade through eBest Investment & Securities Co.'s server. Daishin Securities Co. logs in through CYBOS plus, but Python logs in directly for xingAPI's server. CYBOS plus receives data in a Request/Reply method, and xingAPI receives data through events. Therefore, it is need to write additional codes to handle the event. Data processing in xingAPI is shown in Figure 2 [14]. The first step is to implement classes for event handling. The second step is to create an instance of the XAQuery class. The third step is to register res file for TR code. Res file is a file that expresses the structure of methods provided by xingAPI. The Res file is downloaded to the directory where xingAPI is installed, such as C:\eBest\xingAPI\Res. TR code means a code for transaction. The fourth step is to set input data. The fifth step is to request data to server. The sixth step is to wait for an event. The seventh step is to receive data from server.



**Figure 2. Data processing in xingAPI**

Figure 3 is the Python program that receives the current price using xingAPI [14]. The pythoncom module in line 2 encapsulates the OLE automation API. OLE is an abbreviation Object Linking & Embedding, and is

a proprietary technology developed by Microsoft that allows embedding and linking to documents and other objects. XASessionEventHandler class on line 3 contains an OnLogin method that can receive results for login. The login\_state variable in the fourth line is a variable that receives the login event result. It is set to 1 on success and 0 on failure. The Onlogin method in line 5 is called when the eBest Investment & Securities Co. server notifies you of the login result. The XAQueryEventHandler T1102 class in the 11th line defines a class that handles the T1102 class that retrieves the current price. Lines from 18 to 20 define login properties. Lines from 21 to 23 try to log in. Lines from 24 to 25 wait for a response of login from the server. Lines from 29 to 32 request the current price of the code "078020" using the T1102 class. Lines 33 to 34 wait for a response of the current price from the server. Lines from 35 to 37 print the company name and current price from the returned values of T1102. .

```
import win32com.client
import pythoncom

class XASessionEventHandler:
    login_state = 0

    def OnLogin(self, code, msg):
        if code == "0000":
            print("Login Success")
            XASessionEventHandler.login_state = 1
        else:
            print("Login Failure")

class XAQueryEventHandlerT1102:
    query_state = 0

    def OnReceiveData(self, code):
        XAQueryEventHandlerT1102.query_state = 1

# -----
# login
# -----
id = "id"
passwd = "passwd"
cert_passwd = "cert"

instXASession = win32com.client.DispatchWithEvents("XA_Session.XASession", XASessionEventHandler)
instXASession.ConnectServer("hts.ebestsec.co.kr", 20001)
instXASession.Login(id, passwd, cert_passwd, 0, 0)

while XASessionEventHandler.login_state == 0:
    pythoncom.PumpWaitingMessages()

# -----
# t1102
# -----
instXAQueryT1102 = win32com.client.DispatchWithEvents("XA_DataSet.XAQuery", XAQueryEventHandlerT1102)
instXAQueryT1102.ResFileName = "C:\\WeBEST\\xingAPI\\Res\\t1102.res"
instXAQueryT1102.SetFieldData("t1102InBlock", "shcode", 0, "078020")
instXAQueryT1102.Request(0)

while XAQueryEventHandlerT1102.query_state == 0:
    pythoncom.PumpWaitingMessages()

name = instXAQueryT1102.GetFieldData("t1102OutBlock", "hname", 0)
price = instXAQueryT1102.GetFieldData("t1102OutBlock", "price", 0)
print(name, price)
```

**Figure 3. Python program of eBest Investment & Securities Co. for the current price**

Comparing the two APIs for receiving the current stock data, CYBOS plus has a simple Python program because the API manages login and communicates in a Request/Reply method. xingAPI does not have login information and sends it to the server, and sends and receives data by events. Therefore, the main two

differences are the login method and the method of sending and receiving data. As for the login method, CYBOS plus has login information. But xingAPI does not have login information. As for the method of sending and receiving data, Cybos Plus sends data by calling the request method, and receives data by calling the reply method. xingAPI sends and receives data in the form of an event. When a client sends data to the server, a client generates an event and the event listener handles the event of sending data. When the server sends data to a client, the server generates an event and the event listener handles the event of sending data. Therefore, the number of xingAPI codes is more than that of CYBOS plus. The numbers of codes to get the current price are 6 and 31 lines. Time is very important in analyzing stock data, because the stock price changes in real time. Therefore it is also important to compare the time to get stock data. Follow-up research is needed for response time.

### 2.3 Comparison Using for Loop Statements

Users who receive stock data using open API want to receive a lot of data. This includes a large number of individual stock data and data over time. The reception of data over time can be easily received by classes from open APIs. Individual stock data is the main concern of users, and it is important to support it smoothly. We use loop statements to implement to receive numbers of individual stock data in Python.

We will compare loop statements to receive individual stock data with two open APIs. Python code that uses a loop statement is replaced by codes that receive two stock data, because codes that receive two data can be easily extended to receive multiple data. Figure 4 is the program that receives two current prices using CYBOS plus. Figure 4 shows that CYBOS plus accepts not only list-type data but also tuple-type data and dictionary-type data, and executes a loop statement. Figure 5 is the result of Figure 4. The two current prices are well received. And we find that CYBOS plus supports the basic commands provided by Python.

```
import win32com.client
instStockMst= win32com.client.Dispatch("dscbo1.StockMst")
# List
codeList = ['A000020', 'A000040']
# Tuple
codeList = ('A000020', 'A000040')
# Dictionary
codeList = {'A000020': 'DongWha Pharm', 'A000040': 'KR Moters'}
for code in codeList :
    instStockMst.SetInputValue(0, code)
    instStockMst.BlockRequest()
    print(code, instStockMst.GetHeaderValue(11))
```

Figure 4. Python program of Daishin Securities Co. for loop of current prices

```
A000020 15300
A000040 932
```

Figure 5. Result of the program of Figure 4

Figure 6 is the program that receives two current prices using xingAPI by T1102. Requested the present price of two stocks, but only received one present price. The results are shown in Figure 7. In the manual, one object can handle only one TR by T1102. In order to receive multiple data by changing the codes in the same TR, we should create as many objects as the number of codes. Therefore, it is not possible to use loop

statements in list type for receiving multiple current prices. Using a loop statement in one TR is to view the result by indexes on the downloaded data at once.

```
# -----
# t1102
# -----
instXAQueryT1102 = win32com.client.DispatchWithEvents("XA_DataSet.XAQuery", XAQueryEventHandlerT1102)
instXAQueryT1102.ResFileName = "C:\\WeBEST\\XingAPI\\Res\\t1102.res"

codeList = ['000225', '000640']

for i in codeList:

    instXAQueryT1102.SetFieldData("t1102InBlock", "shcode", 0, i)
    instXAQueryT1102.Request(0)

    while XAQueryEventHandlerT1102.query_state == 0:
        pythoncom.PumpWaitingMessages()

    count = instXAQueryT1102.GetBlockCount("t1102OutBlock")

    shcode = instXAQueryT1102.GetFieldData("t1102OutBlock", "shcode", 0)
    price = instXAQueryT1102.GetFieldData("t1102OutBlock", "price", 0)
    print(shcode, price)
```

Figure 6. Python program of eBest Investment & Securities Co. for loop of current prices

000225 11200

Figure 7. Result of the program of Figure 6

Figure 8 shows the log for the program of Figure 6. We can see that DevCenter receives as many current prices as requested by T1102. However, there is no class or method that can show the second data. Therefore, it explains that one object handles only one TR.

SR	TRCode	Pos	Length	Cont	MsgCode	BlockName	Data	Time
1TD	t1102	L	7	0		000225		10:50:31...
1MS		L	13			00000:		10:50:31...
1TD	t1102	L	1516	0		t1102Ou...	?0013300?0001850?16.16?000000046867 00011450?000...	10:50:31...
1MS		L	13			00000		10:50:31...
1RL	t1102	L	0					10:50:31...
1TD	t1102	L	7	0		000640		10:50:31...
1TD	t1102	L	1516	0		t1102Ou...	?0105500?0001000?00.94?000000005037 00106500?0...	10:50:31...
1MS		L	13			00000:		10:50:31...
1RL	t1102	L	0					10:50:31...

Figure 8. Log of DevCenter for the program of Figure 6

### 3. Conclusion

Securities and investment services had the most data per company on the average, and used the most data, which say systems for large data should be introduced in the securities and investment fields. Investors are increasingly demanding to invest through their own analysis methods. Therefore, securities and investment companies provide stock-related data to investors through open API. The data received using the open API is in text format. Python is effective and convenient for requesting and receiving text data.

We investigate there are 22 major securities and investment companies in Korea and only 6 companies such as Daishin Securities Co. and eBest Investment & Securities Co., Kiwoom Securities Co., Yuanta

Securities Co., Hana Financial Investment Co., IBK Securities Co., support open API. Only Daishin Securities Co. supports Python officially.

We compare how to receive stock data through open API using Python, and Python programming features. The open APIs for the study are Daishin Securities Co. and eBest Investment & Securities Co., because Daishin Securities Co. is the only open API that officially supports Python, and eBest Investment & Securities Co. can represent open API that unofficially supports Python. Comparing the two APIs for receiving the current stock data, the main two differences are the login method and the method of sending and receiving data. As for the login method, CYBOS plus has login information. But xingAPI does not have login information. As for the method of sending and receiving data, Cybos Plus sends and receives data by calling the request method, and the reply method. xingAPI sends and receives data in the form of an event. Therefore, the number of xingAPI codes is more than that of CYBOS plus. The numbers of codes to get the current price are 6 and 31 lines.

Comparing for loop statements, CYBOS plus executes a loop statement by lists and tuple, dictionary. And we find that CYBOS plus supports the basic commands provided by Python. We requested the present price of two stocks by xingAPI with T1102, but only received one present price. In order to receive multiple data by changing the codes in the same TR, we should create as many objects as the number of codes. Therefore, it is not possible to use loop statements in list for receiving multiple objects.

Time is very important in analyzing stock data, because the stock price changes in real time. Therefore it is also important to compare the time to get stock data. Follow-up research is needed for response time.

## Acknowledgement

This research was supported by Seokyeong University in 2020.

## References

- [1] KB management Institute, Understanding big data and implications for financial services, 2012.
- [2] McKinsey Global Institute, Big Data: The next frontier for innovation, competition, and productivity, 2011.
- [3] Chun, Se-Hak, "The Effect of Data Size on the k-NN Predictability: Application to Samsung Electronics Stock Market Prediction," *JIntell Inform Syst(jiis)*, Vol. 25, No. 3, pp. 239~251, 2019.  
DOI: <http://dx.doi.org/10.13088/jiis.2019.25.3.239>
- [4] Jinho Byun, "Current Status and Perspectives of Fintech Innovation," *Journal of New Industry and Business*, Vol 51, No. 2, pp. 35-48, 2018  
DOI: 10.30753/emr.2018.36.2.003
- [5] Seonghoon Jang and Seung-Jung Shin, "Machine Learning-Based Programming Analysis Model Proposal: Based on User Behavioral Analysis," *International Journal of Advanced Smart Convergence(IJASC)*, Vol.9, No.4, pp. 179-183, 2020.  
DOI: <http://dx.doi.org/10.7236/IJASC.2020.9.4.179>
- [6] Soonduck Yoo, "Research on the Methodology for Policy Deriving to active Artificial Intelligence," *The Journal of The Institute of Internet, Broadcasting and Communication(JIIBC)*, Vol. 20, No. 5, pp.187-193, 2020.  
DOI: <https://doi.org/10.7236/JIIBC.2020.20.5.187>
- [7] Gui Yeol Ryu, "A Study on Open API of Securities and Investment Companies for Activating Big Data," *International Journal of Advanced Smart Convergence (IJASC)*, Vol. 8, No. 2, pp. 102-108, 2019.  
DOI: <http://dx.doi.org/10.7236/IJASC.2019.8.2.102>
- [8] Daishin Securities Co. <https://www.daishin.com>.
- [9] eBest Investment & Securities Co. <https://www.ebestsec.co.kr>.
- [10] Kiwoom Securities Co. <https://www.kiwoom.com>.
- [11] Yuanta Securities Co. <https://www.myasset.com>.
- [12] Hana Financial Investment Co. <https://www.hanaw.com>.
- [13] IBK Securities Co. <https://www.ibk.com>.
- [14] Dae Pyo Jo, Algorithm Trading Learning in Python, Wikibooks, 2019.