Data is the future

Wind API Manual

2020.09.16

Win.d

Catalog

1 Overview of Wind API

Wind API supports six programming languages: Python, Matlab, C#, C++, R, and VBA. API provides investors with historical daily, tick, minute and real-time data of stock, bond, fund, commodity, index, foreign exchange, and option, as well as the special statistical reports of all varieties of the Chinese market and the data of China and overseas stocks, macro data. API users can extract data through API functions and conduct in-depth research in the fields of financial engineering, intelligent investment advisory, strategy backtesting, simulated trading, FICC, multi-factor stock selection, machine learning, etc.

1.1 Advantages of Wind API

First, users have flexible choice of six languages. Researchers who are proficient in any programming language of Python, Matlab, C#, C++, R and VBA can choose the appropriate language to write according to the strategy needs.

Second, API functions are powerful and easy to use. Provide a variety of data acquisition functions, portfolio functions, position placing functions and auxiliary date functions, simple and fast extraction of required data, to meet the needs of quantitative investment in all directions.

Third, data is very complete. API covers all types of stocks, bonds, commodities, funds, indices, foreign exchange, options in the Chinese and global markets. API provides daily and tick, minute, real-time financial data, as well as data such as global macroeconomic and global enterprise library, to meet different users' need.

2 Introduction of Wind API Function

Wind API functions' variety are rich, providing more than ten functions to use, including providing market data functions, portfolio functions, trading functions and auxiliary date functions, etc., which fully meet the needs of data acquisition and quantitative investment.

2.1 Data Function

2.1.1 WSD: Time-series Data on the Daily Basis

WSD is used to get time-series data on the daily basis of selected securities, including market information, fundamental information, and security analysis, etc. WSD supports acquiring time-series data of a single indicator of multiple securities and multiple indicators of a single security.

- Function model: w.wsd(codes, fields, beginTime, endTime, options)
- Parameters

Parameters	Type	Optional	Default Value	Description
codes	str/list	No	N.A.	Security tickers. This parameter accepts both a single ticker and multiple tickers, e.g. "600030.SH" and ["600010.SH", "000001.SZ"]
fields	str/list	No	N.A.	A list of indicators. This parameter supports acquiring both a single indictor and multiple indicators, e.g. "CLOSE,HIGH,LOW,OPEN"
beginTime	str/datetime	Yes	endTime	Begin date. The parameter accepts various formats, e.g. "2016-01-01", "20160101", "2016/01/01", "-5D"(five trading days before the current date)
endTime	str/datetime	Yes	Current system date	End date. The parameter accepts various formats, e.g. "2016-01-05", "20160105", "2016/01/05", "-2D" (two trading days before the current date)
options	str	Yes	“ ”	Optional parameter. Please refer to Code Generator and API Help Center.

2.1.2 WSS: Daily Cross-Sectional Data

WSS is used to get cross-sectional data for a historical day of selected securities, including market information, fundamental information, and security analysis, etc. WSS supports acquiring cross-sectional data of multiple indicators of multiple securities.

- Function model: w.wss(codes, fields, options)
- Parameters

Parameters	Type	Optional	Default Value	Description
codes	str/list	No	N.A.	Security tickers. This parameter accepts both a single ticker and multiple tickers, e.g. "600030.SH" and ["600010.SH", "000001.SZ"]
fields	str/list	No	N.A.	A list of indicators. This parameter supports acquiring both a single indictor and multiple indicators, e.g. "CLOSE,HIGH,LOW,OPEN"
options	str	Yes	“ ”	Optional parameter. Please refer to Code Generator and API Help Center.

2.1.3 WST: Intraday Tick Data

WST is used to get handicap snapshot data at five frequencies and tick data of the selected security.

- Function model: w.wst(codes, fields, beginTime, endTime, options)
- Parameters

Parameters	Type	Optional	Default Value	Description
codes	str/list	No	N.A.	Security id. This parameter accepts a single security, e.g. "600030.SH"
fields	str/list	No	N.A.	A list of indicators. This parameter supports acquiring both a single indictor and multiple indicators, e.g. "CLOSE,HIGH,LOW,OPEN"
beginTime	str/datetime	Yes	endTime	Begin time for minute-resolution data. e.g. "2016-01-01 09:00:00"
endTime	str/datetime	Yes	Current system date	End time for minute-resolution data. e.g. "2016-01-01 09:00:00"
options	str	Yes	“ ”	Optional parameter. Please refer to Code Generator and API Help Center.

2.1.4 WSI: Minute-Line Data

WSI is used to get minute-line data of selected securities. The bar size can be set to 1, 3,5,10,15,30,60 minutes.

- Function model: w.wsi(codes, fields, beginTime, endTime, options)
- Parameters

Parameters	Type	Optional	Default Value	Description
codes	str/list	No	N.A.	Security tickers. This parameter accepts both a single ticker and multiple tickers, e.g. "600030.SH" and ["600010.SH", "000001.SZ"]
fields	str/list	No	N.A.	A list of indicators. This parameter supports acquiring both a single indictor and multiple indicators, e.g. "CLOSE,HIGH,LOW,OPEN"
beginTime	str/datetime	Yes	endTime	Begin time for minute-resolution data. e.g. "2016-01-01 09:00:00"
endTime	str/datetime	Yes	Current system date	End time for minute-resolution data. e.g. "2016-01-01 09:00:00"
options	str	Yes	“ ”	Optional parameter. Please refer to Code Generator and API Help Center.

2.1.5 WSQ: Real-Time Market Quotation Data

WSQ is used to get real-time market quotation data of selected securities. It can be used to request one-off real-time snapshot data or get real-time data via subscription.

- Function model: w.wsq(codes, fields, options, func)
- Parameters

Parameters	Type	Optional	Default Value	Description
codes	str/list	No	N.A.	Security tickers. This parameter accepts both a single ticker and multiple tickers, e.g. "600030.SH" and ["600010.SH", "000001.SZ"]
fields	str/list	No	N.A.	A list of indicators. This parameter supports acquiring both a single indictor and multiple indicators, e.g. "CLOSE,HIGH,LOW,OPEN"
options	str	Yes	“ ”	Optional parameter. Please refer to Code Generator and API Help Center.

func	str	Yes	None	The default value for func is None. With the default value, the function loads one-off snapshot data. If func=DemoWSQCallback, the function returns market data via subscription. Please see the examples for how to define DemoWSQCallback in API Help Center.

2.1.6 WSES: Time-series Data of Sectors on Daily Basis

WSES is used to download historical time-series data of the sectors of Shanghai, Shenzhen, Hong Kong and global markets, including the daily market data, fundamental data and predicted profits based on the stock-level data of each sector.

- Function model: w.wses(codes, fields, beginTime, endTime, options)
- Parameters

Parameters	Type	Optional	Default Value	Description
codes	str/list	No	N.A.	Sector tickers. This parameter accepts both a single ticker and multiple tickers, e.g. : "a001010100", ["a001010200","a001010200"]
fields	str/list	No	N.A.	This parameter only accepts a single indicators "sec_close_avg"
beginTime	str/datetime	Yes	endTime	Begin date. The parameter accepts various formats, e.g. "2016-01-01", "20160101", "2016/01/01", "-5D"(five trading days before the current date)
endTime	str/datetime	Yes	Current system date	End date. The parameter accepts various formats, e.g. "2016-01-05", "20160105", "2016/01/05", "-2D" (two trading days before the current date)
options	str	Yes	“ ”	Optional parameter. Please refer to Code Generator and API Help Center.

2.1.7 WSEE: Daily Cross-section Data of Sectors

WSEE is used to download historical daily cross-section data of sectors in Shanghai, Shenzhen, Hong Kong and global markets, e.g. the average market data and the average

financial data of all A-share sectors.

- Function model: w.wsee(codes, fields, options)
- Parameters

Parameters	Type	Optional	Default Value	Description
codes	str/list	No	N.A.	Sector tickers. This parameter accepts both a single ticker and multiple tickers, e.g. : "a001010100", ["a001010200","a001010200"]
fields	str/list	No	N.A.	This parameter only accepts a single indicators "sec_close_avg"
options	str	Yes	“ ”	Optional parameter. Please refer to Code Generator and API Help Center.

2.1.8 WSET: Get Reports

WSET is used to get reports for the details of various datasets, covering sector constituents, index constituents, ETF purchasing and redeeming, split capital funds, margin trading underlying, short selling underlying, collateral of margin trading and short selling, buy-back collateral, stocks of trading suspension, stocks of trading resumption as well as cash and stock dividend, etc.

- Function model: w.wset(tableName, options)
- Parameters

Parameters	Type	Optional	Default Value	Description
tableName	str	No	N.A.	Generate the tablename, within Code Generator, e.g. "SectorConstituent"
options	str	Yes	“ ”	Optional parameter. Please refer to Code Generator and API Help Center.

2.1.9 EDB: Global Macroeconomic Data

EDB is used to load data from Wind Macroeconomic Database, providing users an easy tool to check and export macroeconomic and industrial data. The macroeconomic database covers macro economy of China, global macro economy, industrial economy, commodity and interest rates.

- Function model: edb(codes, beginTime, endTime, options)

- Parameters

Parameters	Type	Optional	Default Value	Description
codes	str/list	No	N.A.	Enter the IDs of the desired indicators or generate the code within Code Generator, e.g. "M5567877,M5567878" ,["M5567877","M5567878"]
beginTime	str/datetime	Yes	endTime	Begin date. The parameter accepts various formats, e.g. "2016-01-01", "20160101", "2016/01/01", "-5D"(five trading days before the current date)
endTime	str/datetime	Yes	Current system date	End date. The parameter accepts various formats, e.g. "2016-01-05", "20160105", "2016/01/05", "-2D" (two trading days before the current date)
options	str	Yes	“ ”	Optional parameter. Please refer to Code Generator and API Help Center.

2.1.10 WND: Historical News Data

WND is used to obtain historical news data of selected securities or industry sector, including news title, publication time, url, abstract, etc.

- Function model: w.wnd(codes, beginTime, endTime, options)
- Parameters

Parameters	Type	Optional	Default Value	Description
codes	str/list	No	N.A.	Security or industry tickers. This parameter accepts both a single ticker and multiple tickers, e.g. "600030.SH" and ["600010.SH", "BK"]
beginTime	str/datetime	Yes	endTime	Begin time for news data. e.g. "2020-01-01 09:00:00"
endTime	str/datetime	Yes	Current system date	End time for news data. e.g. "2020-01-01 09:00:00"
options	str	Yes	“ ”	Optional parameter. Please refer to Code Generator and API Help Center.

- Parameters Aggregated within options

Parameters	Type	Optional	Default Value	Description
field	str	No	N.A.	You can filter the news field you want to return by the “field” parameters, which is selected all by default. The meaning of parameters is as follows: id: news id title: news title time: news publication time url: news URL link address source: news source abstract: news abstract relevant_windcodes: all windcodes related to this news important: is it an important news

2.1.11 WNQ: Real-Time News Data

WNQ is used to obtain real-time news data of selected securities or industry sector, including news title, publication time, url, abstract, etc. It can be used to request one-off real-time snapshot data or get real-time news data via subscription.

- Function model: w.wnq(codes, options, func)
- Parameters

Parameters	Type	Optional	Default Value	Description
codes	str/list	No	N.A.	Security or industry tickers. This parameter accepts both a single ticker and multiple tickers, e.g. "600030.SH" and ["600010.SH", "BK"]
options	str	Yes	“ ”	Optional parameter. Please refer to Code Generator and API Help Center.
func	str	Yes	None	The default value for func is None. With the default value, the function loads one-off snapshot data. If func=DemoCallback, the function returns news data via subscription.

2.1.12 WNC: News Content Data

WNC is used to obtain news content of selected newsID. By entering the newsID, you can extract the news content corresponding to the ID.

- Function model: w.wnc(newsID, options)
- Parameters

Parameters	Type	Optional	Default Value	Description
newsID	str	No	N.A.	Only one newsID can be entered at a time, such as “503289717”. The newsID can be obtained through WND/WNQ function and each news has a unique newsID.
options	str	Yes	“ ”	Optional parameter. Please refer to Code Generator and API Help Center.

- Return: data returned in JSON format, with a total of 9 items: ID, title, time, url, source, abstract, relevant_windcodes, important, and content.

2.1.13 WGEL: Enterprise Library API

WGEL is used to get data information in the Wind Global Enterprise Library. For details, please refer to the documentation of Enterprise Library API on WFT.

Keyboard Wizard shortcut: EAPI

2.1.14 WEQS: Get Securities Screening Results

WEQS is used to get a screening scheme in the Equity Screener, Fund Screener, and Bond Screener models on WFT.

- Function model: w.weqs(filtername)
- “filtername” is the scheme name of Equity Screener, Fund Screener, and Bond Screener models.

2.2 Trading Function

Wind API supports the mock trading including equity, futures, option or real futures trading. Wind API trade interface contains five functions: tlogon(log on), tlogout(log out), torder(submit orders), tcancel(withdraw orders), tquery(query).

2.2.1 tlogon: Log on a Trading Account

tlogon is used to log on capital account or quantitative mock account. After successfully logging on, the system will generate a logon ID, which is used as a unique ID for the logon.

- Function model: w.tlogon(BrokerID, DepartmentID, LogonAccount, Password, AccountType, options, func)
- Parameters

Parameters	Type	Optional	Default Value	Description
BrokerID	str	No	N.A.	Enter the broker ID for the mock trading or real futures trading.
DepartmentID	str	No	N.A.	Enter the broker's Branch ID for mock trading account or real trading account.
LogonAccount	str	No	N.A.	Enter the capital account number for the mocking trading or real trading.
Password	str	No	N.A.	Enter the password for the capital account. If it is a mock trading, enter any password.
AccountType	str	No	N.A.	"SHSZ": Shenzhen and Shanghai A shares, "CZC": Zhengzhou Commodity Exchange, "SHF": Shanghai Future Exchange, "DCE": Dalian Commodity Exchange, "CFE": China Financial Futures Exchange, "SHO": Shanghai Stock Exchange Option, "HK": Hong Kong Stock Exchange
options	str	Yes	" "	Optional parameter. Please refer to Code Generator and API Help Center.
func	str	Yes	None	Callback function to input CTP Order/Transaction. Log on to CTP account for futures and use callback function to return the information of orders and transactions. 001: order, 4002: transaction

2.2.2 tlogout: Log out a Trading Account

tlogout is used to log out the trading system. Log out with the logonID returned by tlogon. To log out, you should also use the logonID.

- Function model: w.tlogout(LogonID, options)
- Parameters

Parameters	Type	Optional	Default Value	Description
LogonID	str	Yes	None	Use the logonID returned by w.tlogon or w.tquery to log out an account. If there is only one account on, don't need to specify the logonID.
options	str	Yes	“ ”	Optional parameter. Please refer to Code Generator and API Help Center.

2.2.3 torder: Submit Orders

torder is used to submit orders in the account, which is logged on.

- Function model: w.torder(SecurityCode, TradeSide, OrderPrice, OrderVolume, options)
- Parameters

Parameters	Type	Optional	Default Value	Description
SecurityCode	str	No	N.A.	Enter the security code, e.g. “600000.SH”. You can also enter transaction code. In this case, you also need to specify MarketType.
TradeSide	str	No	N.A.	Set the direction of trading according to the type of instruments. Please refer to Code Generator and API Help Center.
OrderPrice	str/int	No	N.A.	Specify the price of order, e.g. "10.12", 10.12
OrderVolume	str/int	No	N.A.	Specify the size of order, e.g. "100", 100
options	str	Yes	“ ”	Optional parameter. Please refer to Code Generator and API Help Center.

- Parameters Aggregated within options

Parameters	Type	Optional	Default Value	Description
OrderType	str	Yes	“LMT”	Specify the type of order. Please refer to Code Generator and API Help Center.
HedgeType	str	Yes	“SPEC”	Specify a type of trading. "SPEC" or "0": speculation; "HEDG" or "1": hedge.
LogonID	str	Yes	N.A.	Specify the logonID returned by tlogon or tquery function to distinguish accounts. You don't need

				to specify LogonID, if there is only one account logged on.
MarketType	str	Yes	“SH”	Set the market, Please refer to Code Generator and API Help Center.

2.2.4 tcancel: Cancel Order

tcancel is used to cancel the specified order.

- Function model: w.tcancel(OrderNumber, options)
- Parameters

Parameters	Type	Optional	Default Value	Description
OrderNumber	str	No	N.A	The order number of the order to cancel, which is generated when the order is submitted and can be inquired via w.tquery("Order") .
options	str	Yes	“ ”	Optional parameter. Please refer to Code Generator and API Help Center.

2.2.5 tquery: Get the information of Trading

tquery is used to request for the information about trading, including inquiring capital, position, order, trader, account, etc.

- Function model: w.tquery(qrycode, options)
- Parameters

Parameters	Type	Optional	Default Value	Description
qrycode	str	No	N.A	Enter the content to query, e.g. "Order". This parameter can be set to the following values: "Capital", "Position", "Order", "Trade", "Account", "LogonID"
options	str	Yes	“ ”	Optional parameter. Please refer to Code Generator and API Help Center.

- Parameters Aggregated within options

Parameters	Type	Optional	Default Value	Description
LogonID	str	Yes	N.A.	Don't need to specify this parameter, If there is only one account logged on.
RequestID	str	Yes	N.A.	The corresponding ID for the request, which is generated by the system. To

				inquiry day order 2/order, you can use the requestID returned by torder , e.g. "RequestID=3"
OrderNumber	str	Yes	N.A.	It is useful if "Qrycode=Order". Order number can be found by w.tquery("Order")
MarketType	str	Yes	“SH”	This parameter is required to specify if SecurityCode is trade code. Please refer to Code Generator and API Help Center.
OrderType	str	Yes	All	The type of the order to inquiry. Therea are two types: All and Withdrawable.
WindCode	str	Yes	N.A.	To request for the information of a specified stock, for example "WindCode=002311.SZ". Qrycode should be 'Order', 'Position' or 'Trade'
BrokerID	str	Yes	N.A.	Query the information of Broker, e.g."BrokerID=0000"

2.3 Portfolio Function

2.3.1 WPF: Portfolio Reports

WPF is used to get the performance report of a portfolio for a period from the Asset Management System (AMS) and Portfolio Management System (PMS).

- Function model: w.wpf(productname, tablename, options)
- Parameters

Parameters	Type	Optional	Default Value	Description
productName	str	No	N.A.	ID or name of a portfolio, which can be found in PMS or AMS module of WFT, e.g. "Global Portfolio Management Demo"
tablename	str	No	N.A.	Name of the indicators in the report, e.g. NetHoldingValue
options	str	Yes	“ ”	Optional parameter. Please refer to Code Generator and API Help Center.

- Parameters Aggregated within options

Parameters	Type	Optional	Default Value	Description
view	str	Yes	N.A.	Select the module where the portfolio is managed, either "AMS" or "PMS"
Owner	str	Yes	N.A.	If view=PMS and the portfolio is shared by others, it is required to enter the Wind account of the owner who created the portfolio, e.g. "Owner=W0817573"
date	str	Yes	N.A.	Date of the cross-section data, e.g. "date = 20180302"
startDate	str	Yes	N.A.	Start date of the desired data, e.g. "startDate = 20180531"
endDate	str	Yes	N.A.	End date of the desired data, e.g. " endDate = 20180731"
Currency	str	Yes	"ORIGINAL"	Currency for the data, supporting "ORIGINAL", "HKD", "USD", "CNY"
sectorcode	str	Yes	N.A.	categorised according to assets or total market value, e.g. "sectorcode=101"
MarketCap	str	Yes	N.A.	It is used to categorized the portfolio according to total market value. It can be set in the following format: " sectorcode=208, MarketCap=1000,500,100,50"
displaymode	str	Yes	N.A.	Select the way to display: 1: details, 2: category and 3: all.

2.3.2 WPS: Multi-dimensional Data of Portfolios

WPS is used to load cross-section data of a PMS or AMS portfolio, including fundamental information, performance and transaction statistics.

- Function model: w.wps(PortfolioName, fields, options)
- Parameters

Parameters	Type	Optional	Default Value	Description
PortfolioName	str	No	N.A.	ID or name of a portfolio, which can be found in PMS or AMS module of WFT, e.g. "Global Portfolio Management Demo"

fields	str	No	N.A.	Name of the indicators in the report, e.g. NetHoldingValue
options	str	Yes	“ ”	Optional parameter. Please refer to Code Generator and API Help Center.

- Parameters Aggregated within options

Parameters	Type	Optional	Default Value	Description
view	str	Yes	N.A.	Select the module where the portfolio is managed, either "AMS" or "PMS"
Owner	str	Yes	N.A.	If view=PMS and the portfolio is shared by others, it is required to enter the Wind account of the owner who created the portfolio, e.g. "Owner=W0817573"
date	str	Yes	N.A.	Date of the cross-section data, e.g. "date = 20180302"
startDate	str	Yes	N.A.	Start date of the desired data, e.g. "startDate = 20180531"
endDate	str	Yes	N.A.	End date of the desired data
Currency	str	Yes	"ORIGINAL"	Currency for the data, supporting "ORIGINAL", "HKD", "USD", "CNY"

2.3.3 WPD: Time-Series Data of a Portfolio

WPD is used to load time-series data of a PMS or AMS portfolio on daily basis, including holdings and performance.

- Function model: w.wpd(PortfolioName, fields, beginTime, endTime, options)
- Parameters

Parameters	Type	Optional	Default Value	Description
PortfolioName	str	No	N.A.	ID or name of a portfolio, which can be found in PMS or AMS module of WFT, e.g. "Global Portfolio Management Demo"
fields	str	No	N.A.	Name of the indicators in the report, e.g. NetHoldingValue
beginTime	str	Yes	endTime	Begin date of the desired data, e.g. "startDate = 20180531"
endTime	str	Yes	Current system date	End date of the desired data, e.g. " endDate = 20180731"

options	str	Yes	“ ”	Optional parameter. Please refer to Code Generator and API Help Center.

- Parameters Aggregated within options

Parameters	Type	Optional	Default Value	Description
view	str	Yes	N.A.	Select the module where the portfolio is managed, either "AMS" or "PMS"
Owner	str	Yes	N.A.	If view=PMS and the portfolio is shared by others, it is required to enter the Wind account of the owner who created the portfolio, e.g. "Owner=W0817573"
Period	str	Yes	“D”	"D": daily, "W": weekly, "M": monthly, "Q": quarterly, "S": semiannually, "Y": annually
Fill	str	Yes	N.A.	Method to fill missing values. "Previous": use the last valid observation, "Blank": return an empty value. To set a specified value, use ShowBlank=X, where X is a customised value.
Currency	str	Yes	"ORIGINAL"	Currency for the data, supporting "ORIGINAL", "HKD", "USD", "CNY"

2.3.4 WUPF: Upload Portfolio

To realize programmed rebalance and implement backtesting, the quantitative platform provides WUPF function to adjust holdings. There are three methods to adjust the position: “Flow Uploading”, “Weight Uploading” and “Position Uploading”.

- Function model:w.wupf(PortfolioName, TradeDate, WindCode, Quantity, CostPrice, options)
- Parameters

Parameters	Type	Optional	Default Value	Description
PortfolioName	str	No	N.A.	ID or name of a portfolio, which can be found in PMS or AMS module of WFT, e.g. "Global Portfolio Management Demo"
TradeDate	str	No	N.A.	The date to adjust position, e.g. "20151231"

WindCode	str	No	N.A.	The securities to adjust position. You can adjust multiple securities by setting this parameter to a list. Cash is treated as a security, of which the number is the amount and the price is 1. Currently, one cash is allowed to upload, e.g. "600000.SH, CNY "
Quantity	str	No	N.A.	The amount of positions to adjust
CostPrice	str	No	N.A.	The price to calculate costs for position adjustment, including commissions and other transaction fees.
options	str	Yes	“ ”	Optional parameter. Please refer to Code Generator and API Help Center.

- Parameters Aggregated within options

Parameters	Type	Optional	Default Value	Description
Owner	str	Yes	N.A.	If the portfolio is shared by others, it is required to enter the Wind account of the owner who created the portfolio, e.g. "Owner=W0817573"
Direction	str	Yes	“Long”	Long or short. It is also valid for future.
HedgeType	str	Yes	“Spec”	Specify a type of trading. (0) "SPEC": speculation, (1) "HEDG": hedge. If choosing "HEDG", you need a special account for hedging
CreditTrading	str	Yes	N.A.	Specify if it is credit trading, e.g. "CreditTrading=No,No,No"
TotalAsset	str	Yes	10000000	Set the initial total asset
Method	str	Yes	N.A.	Method to adjust positions. "BuySell": buy or sell (change cash as well), "InOut": in or out (won't change cash)
AssetType	str	Yes	N.A.	AssetType: "Margin", "Cash", "Equity", "Bond", "Repo", "Fund", "Cmdty", "SFP" (security manager financial products), "Trust", "BFP" (bank financial products), "Pfund". The backend can determine the type of asset; therefore don't need to specify this parameter unless it is "Margin". If AssetType is specified, the backend won't do AssetType checking any more.

type	str	Yes	Position Uploading	The method to adjust the position. The default is position uploading. Set it to "flow" to choose flow uploading. To choose weight uploading, set the parameter to "weight"

- **Flow Uploading**

In this mode, you can adjust the portfolio by either adjusting cash or adjusting position.

1. Adjust Cash. By 'Adjust Cash', you can adjust the amount of cash in the portfolio. If the change in cash is positive, you increase the amount of cash in the portfolio; otherwise, you reduce the amount of cash. Additionally, you can also choose the currency to change.

2. Adjust Position. There are two ways to adjust position, i.e. adjust by trading and asset transfer, among which adjust by trading will change the amount of cash, while asset transfer won't.

- **Weight Uploading**

"Weight Uploading" uploads the positions according to the weights of each security. The uploaded positions become the latest position of the portfolio. The minimum volume to adjust is 1 share or 1 hand.

1. The portfolio is empty before uploading the weights for the first time. If you don't specify the total assets when uploading the portfolio, the default total asset is 10000000; otherwise, the total asset is the specified value.

2. If the portfolio is not empty when uploading the weights, you don't need to specify the total asset.

- **Position Uploading**

"Position Uploading" uploads the positions, including the position of cash, on the rebalancing day. The uploaded position is, therefore, the latest position of the portfolio. "Position Uploading" doesn't have memory to historical position. There are two types of "Position Uploading", i.e. "Adjust Position" and "Adjust Cash".

1. Adjust Cash: adjust the amount of cash in the portfolio and specify the currency to adjust.

2. Adjust Position: upload the positions on the rebalancing day.

- **Reset Portfolio**

To remove all information of positions and capital of a portfolio, set up w.wupf as the following: w.wupf(PortfolioName, "", "", "", "reset=true")

2.4 Date Function

2.4.1 tdays: Get Date Series

tdays is used to get the series of dates in a range with some pre-defined rule.

- Function model: w.tdays(beginTime , endTime, options)
- Parameters

Parameters	Type	Optional	Default Value	Description
beginTime	str	Yes	endTime	Start date of the series of dates. It supports date macro.
endTime	str	Yes	Current system date	Start date of the series of dates. It supports date macro.
options	str	Yes	“ ”	Optional parameter. Please refer to Code Generator and API Help Center.

- Parameters Aggregated within options

Parameters	Type	Optional	Default Value	Description
Days	str	Yes	“Trading”	"Weekdays": weekdays, "Alldays": calendar days, "Trading": trading days.
Period	str	Yes	“D”	D": daily, "W": weekly, "M": monthly, "Q": quarterly, "S": semiannually, "Y": annually
TradingCalendar	str	Yes	“SSE ”	Specify an exchange to get trading calendar. The default is "SSE", Shanghai Stock Exchange.

2.4.2 tdaysoffset: Offset Dates

tdaysoffset is used to get a date shifting the benchmark date by a desired period.

- Function model: w.tdaysOffset(offset, beginTime, options)
- Parameters

Parameters	Type	Optional	Default Value	Description
offset	int	No	N.A.	Number of periods to shift. If offset is positive, add extra periods; vice versa.
beginTime	str	Yes	Current system date	Start date of the series of dates. It supports date macro.
options	str	Yes	“ ”	Optional parameter. Please refer to Code Generator and API Help Center.

- Parameters Aggregated within options

Parameters	Type	Optional	Default Value	Description
Days	str	Yes	“Trading”	"Weekdays": weekdays, "Alldays": calendar days, "Trading": trading days.
Period	str	Yes	“D”	"D": daily, "W": weekly, "M": monthly, "Q": quarterly, "S": semiannually, "Y": annually
TradingCalendar	str	Yes	“SSE ”	Specify an exchange to get trading calendar. The default is "SSE", Shanghai Stock Exchange.

2.4.3 tdayscount: Count the Days between Dates

tdayscount counts the days of a specified date type between two dates.

- Function model: w.tdayscount(beginTime, endTime, options)
- Parameters

Parameters	Type	Optional	Default Value	Description
beginTime	str	Yes	endTime	Start date of the series of dates. It supports date macro.
endTime	str	Yes	Current system date	End date of the series of dates. It supports date macro.
options	str	Yes	“ ”	Optional parameter. Please refer to Code Generator and API Help Center.

- Parameters Aggregated within options

Parameters	Type	Optional	Default Value	Description
Days	str	Yes	“Trading”	"Weekdays": weekdays,"Alldays": calendar days, "Trading": trading days.
TradingCalendar	str	Yes	“SSE ”	Specify an exchange to get trading calendar. The default is "SSE", Shanghai Stock Exchange.

2.4.4 Description for Date Macros

2.4.4.1 General Date Macro

General Date Macro supports relative date expressions, including 'TD' (trading days), 'D' (calendar days), 'W': calendar week, ‘M’: calendar month, 'Q': calendar quarter, 'S': calendar semi year, 'Y': calendar year.

- "-" represent forward shift, while the number specify the periods, which must be an integer. For example, "-5D" means five calendar days before the date of today.
- If "ED", end date, is " ", i.e. blank value, use the current system date.
- Date macros supports addition and subtraction, e.g. "ED-10d", which means ten days before the end date.

- Examples:

1. Set the StartDate to a month ago and EndDate the present date:StartDate="-1M", EndDate=""
2. Set the StartDate to ten trading days ago and EndDate five trading days ago: StartDate="-10TD", EndDate="-5TD".

2.4.4.2 Special Date Macros

There are many Date Macros in Equity Screener and Equity Data Explorer, and API supports those Date Macros. Organized as follows:

Names of Macros	Symbol	Names of Macros	Symbol
End Date	ED	First Quarter of the Current Year	RQ1
Start Date	SD	Second Quarter of the Current Year	RQ2
First Quarter of the Last Year	LQ1	Third Quarter of the Current Year	RQ3
Second Quarter of the Last Year	LQ2	Latest Period	MRQ
Third Quarter of the Last Year	LQ3	First Day of the Current Year	RYF
Annual Report of the Last Year	LYR	First Day of the Second Half of the Year	RHYF
First Day of the Current Month	RMF	Monday of the Current Week	RWF
Last Weekend	LWE	End of Last Month	LME
End of the First Half of the Year	LHYE	End of Last Year	LYE
Date of IPO	IPO	--	--

3 Auxiliary Tools of Wind API

3.1 API Help Center

In order to facilitate users to learn the Wind API, we provide detailed user manuals in six programming languages to help users quickly get started with Wind API. Before using the Wind API, it is recommended to read the API Help Center completely.

The path to open API Help Center: WFT → Quant → API → API Help Center.

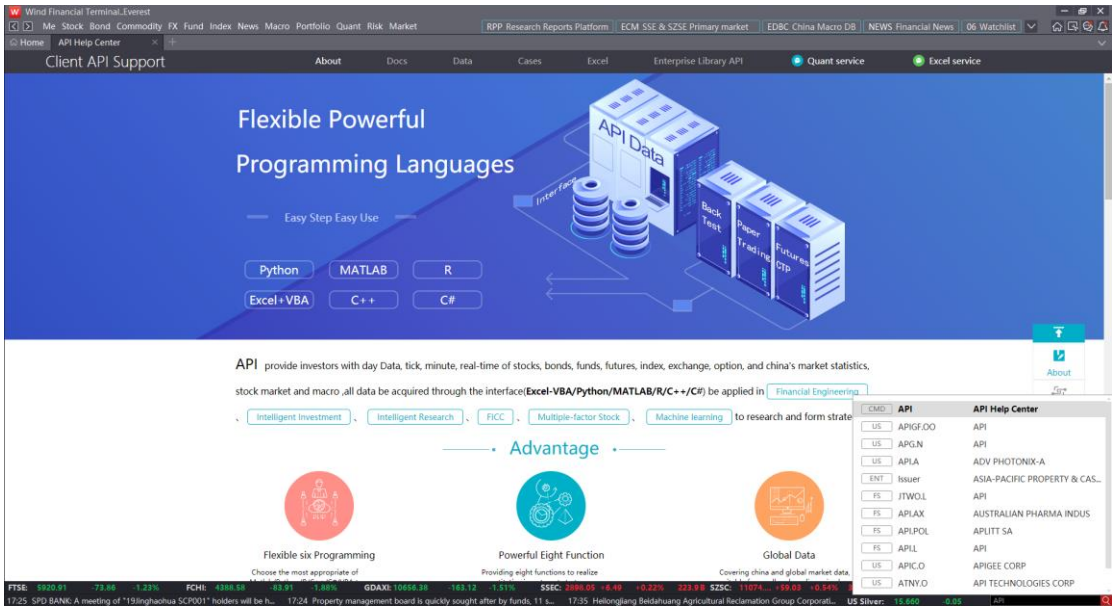
Keyboard Wizard shortcut: API

3.1.1 API Help Center Reading Process

Step 1: API Help Center → Manual → Read Me First. The content in "Read Me First" applies to all programming languages, and explains the core content of the API. No matter what language you use, it is recommended to understand this document before reading the user manuals for each programming language.

Step 2: API Help Center → Manual → API Handbook corresponding to the programming language used. "API Handbook" elaborated how to install API, call API and use functions on a certain programming language. When encountering problems in using functions, you can also view the API Handbook.

Step 3: API Help Center → Cases → Cases corresponding to the programming language used. The API Help Center provides practical cases for six programming languages, such as obtaining all A-share constituents, subscribing real-time data, and retrieve minute data. Users can furtherly develop other functions on the basis of API cases.

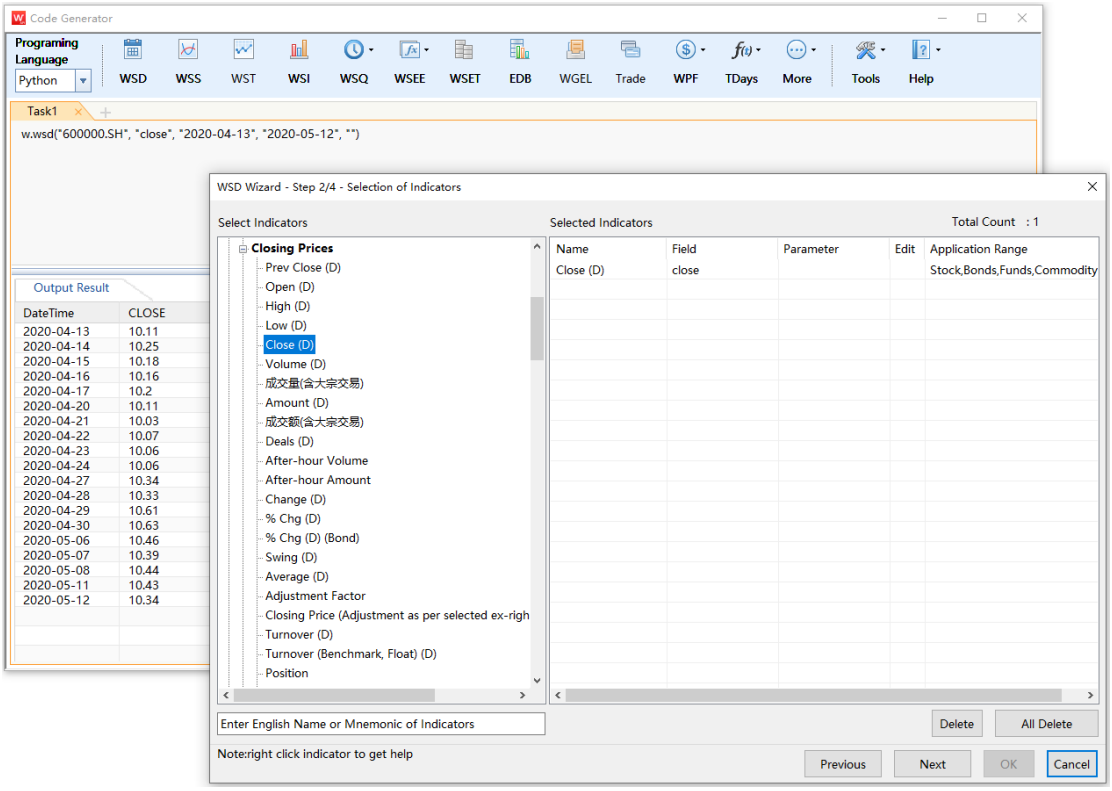


3.2 Code Generator

In order to assist users to generate function codes, Wind API provides a powerful auxiliary tool-Code Generator. Through the Code Generator, you can intuitively see the full picture of the API and conveniently view the indicators' help. With the help of navigation guidance, step-by-step selection and filling in parameters, you can generate the function codes for obtaining data, avoiding the trouble of querying the manual. While generating the function codes, you can also directly display the data result, improving user's efficiency. The Code Generator is built in Matlab and R language, and when using other languages, you can open the Code Generator as an auxiliary tool in the WFT.

How to open the code generator: WTF → quant → API → Code Generator.

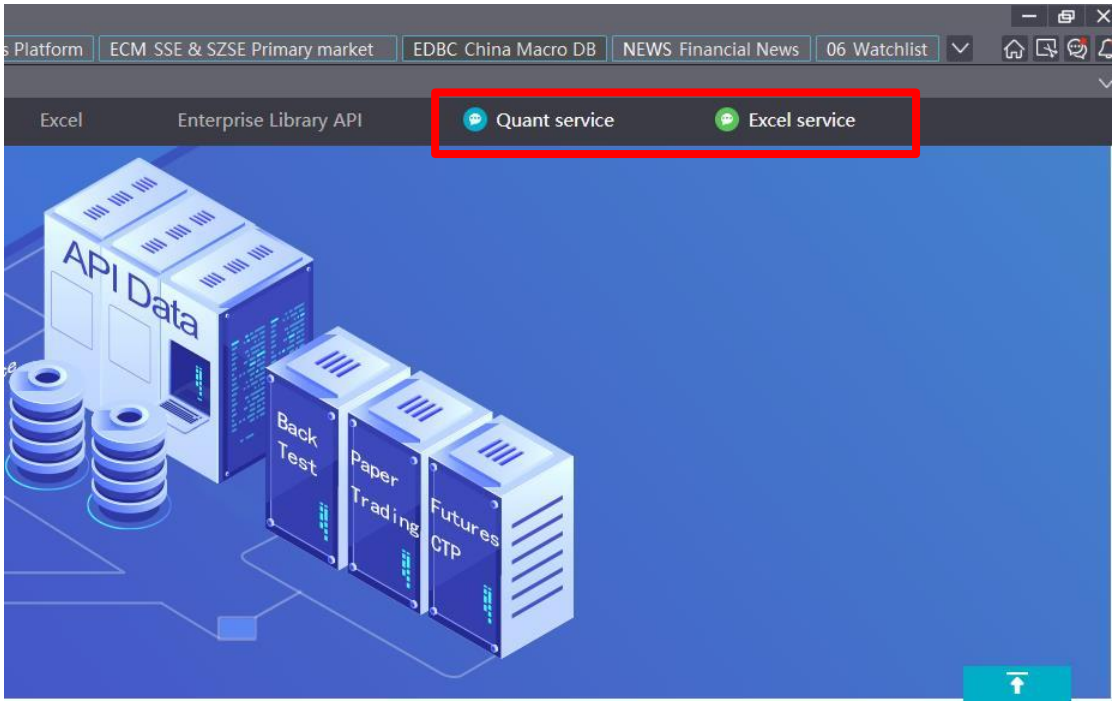
Keyboard Wizard shortcut: CG



3.3 Q&A Group in Wind Message

Wind API team specifically creates API quant Q&A group and Excel add-in Q&A group in Wind Message. Group name: "China Quant Trading Group", group number: 59289; group name: "Excel add-in online communication group", group number: 107768.

You can join the Q & A group by clicking on the upper right corner of the API Help Center or searching for the group number in WM.


Platform
ECM SSE & SZSE Primary market
EDBC China Macro DB
NEWS Financial News
06 Watchlist
Excel
Enterprise Library API
Quant service
Excel service
API Data
Back Test
Paper Trading
Futures CTP

In virtue of specific privileged information in relation to technology and business practice contained in this document, all materials herein shall be kept strictly confidential. As an authorized client of our company, you shall make promises not to disclose any information herein to any third.

Wind Information Co., Ltd.
Address: 7/F Wind Plaza, No.1500 Puming Road, Shanghai, China
Tel: (8621) 2070 0800
Fax: (8621) 2070 0888
Email: sales@wind.com.cn
Http: //www.wind.com.cn