

---

# **xbbg Documentation**

***Release 0.7.7a2***

**Alpha x1**

**Dec 20, 2021**



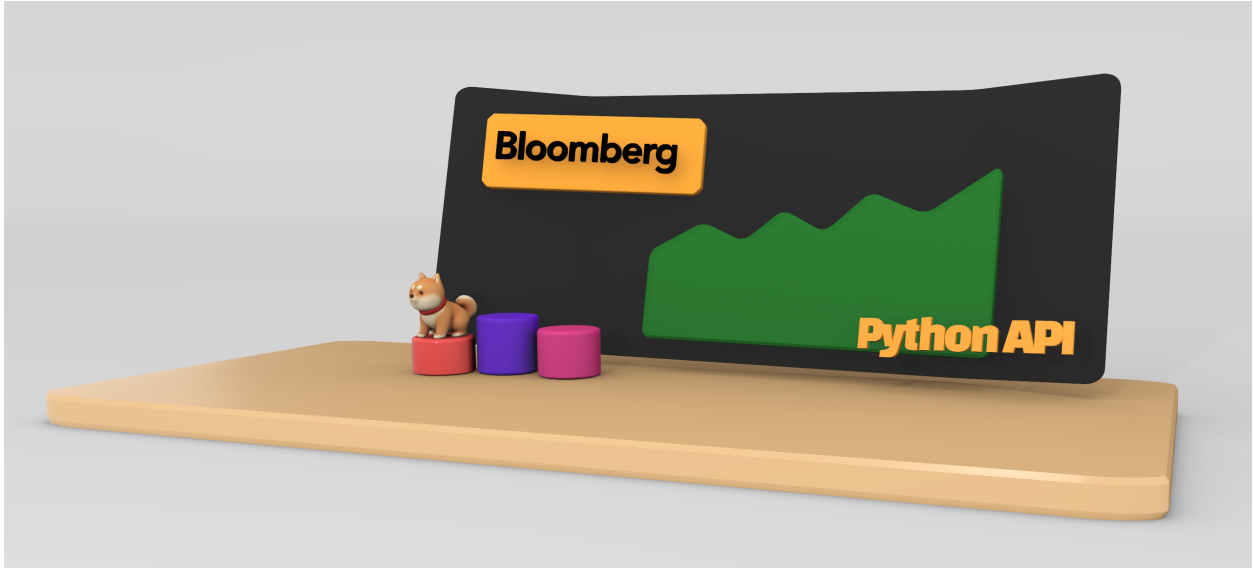
---

## Contents

---

<b>1</b>	<b>Features</b>	<b>3</b>
<b>2</b>	<b>Requirements</b>	<b>5</b>
<b>3</b>	<b>Installation</b>	<b>7</b>
<b>4</b>	<b>What's New</b>	<b>9</b>
<b>5</b>	<b>Tutorial</b>	<b>11</b>
5.1	Basics . . . . .	11
5.2	Data Storage . . . . .	14





Intuitive Bloomberg data API





# CHAPTER 1

---

## Features

---

Below are main features. Jupyter notebook examples can be found [here](#).

- Excel compatible inputs
- Straightforward intraday bar requests
- Subscriptions





## CHAPTER 2

---

### Requirements

---

- Bloomberg C++ SDK version 3.12.1 or higher
  - Visit [Bloomberg API Library](#) and download C++ Supported Release
  - In the `bin` folder of downloaded zip file, copy `blpapi3_32.dll` and `blpapi3_64.dll` to Bloomberg BLPAPI\_ROOT folder (usually `blp/DAPI`)

- Bloomberg official Python API:

```
pip install blpapi --index-url=https://bcms.bloomberg.com/pip/simple/
```

- `numpy`, `pandas`, `ruamel.yaml` and `pyarrow`



## CHAPTER 3

---

### Installation

---

```
pip install xbbg
```



## CHAPTER 4

---

### What's New

---

*0.7.7a2* - Custom *config* and etc. for reference exchange (author *hceh*)

*0.7.6a2* - Use *blp.connect* for alternative Bloomberg connection (author *anxl2008*)

*0.7.2* - Use *async* for live data feeds

*0.7.0* - *bdh* preserves columns orders (both tickers and *flds*). *timeout* argument is available for all queries - *bdtick* usually takes longer to respond - can use *timeout=1000* for example if keep getting empty *DataFrame*.

*0.6.6* - Add flexibility to use reference exchange as market hour definition (so that it's not necessary to add *.yaml* for new tickers, provided that the exchange was defined in */xbbg/markets/exch.yaml*). See example of *bdib* below for more details.

*0.6.0* - Speed improvements and tick data availability

*0.5.0* - Rewritten library to add subscription, BEQS, simplify interface and remove dependency of *pdblp*

*0.1.22* - Remove PyYAML dependency due to security vulnerability

*0.1.17* - Add *adjust* argument in *bdh* for easier dividend / split adjustments



```
In [1]: from xbbg import blp
```

## 5.1 Basics

BDP example:

```
In [2]: blp.bdp(tickers='NVDA US Equity', flds=['Security_Name', 'GICS_Sector_Name'])
Out[2]:
```

	security_name	gics_sector_name
NVDA US Equity	NVIDIA Corp	Information Technology

BDP with overrides:

```
In [3]: blp.bdp('AAPL US Equity', 'Egy_Weighted_Avg_Px', VWAP_Dt='20181224')
Out[3]:
```

	egy_weighted_avg_px
AAPL US Equity	148.75

BDH example:

```
In [4]: blp.bdh(
...:     tickers='SPX Index', flds=['High', 'Low', 'Last_Price'],
...:     start_date='2018-10-10', end_date='2018-10-20',
...: )
Out[4]:
```

	SPX Index		
	High	Low	Last_Price
2018-10-10	2,874.02	2,784.86	2,785.68
2018-10-11	2,795.14	2,710.51	2,728.37
2018-10-12	2,775.77	2,729.44	2,767.13
2018-10-15	2,775.99	2,749.03	2,750.79

(continues on next page)

(continued from previous page)

```

2018-10-16  2,813.46  2,766.91  2,809.92
2018-10-17  2,816.94  2,781.81  2,809.21
2018-10-18  2,806.04  2,755.18  2,768.78
2018-10-19  2,797.77  2,760.27  2,767.78

```

BDH example with Excel compatible inputs:

```

In [5]: blp.bdh(
...:     tickers='SHCOMP Index', flds=['High', 'Low', 'Last_Price'],
...:     start_date='2018-09-26', end_date='2018-10-20',
...:     Per='W', Fill='P', Days='A',
...: )
Out[5]:
      SHCOMP Index
           High      Low Last_Price
2018-09-28    2,827.34  2,771.16    2,821.35
2018-10-05    2,827.34  2,771.16    2,821.35
2018-10-12    2,771.94  2,536.66    2,606.91
2018-10-19    2,611.97  2,449.20    2,550.47

```

BDH without adjustment for dividends and splits:

```

In [6]: blp.bdh(
...:     'AAPL US Equity', 'Px_Last', '20140605', '20140610',
...:     CshAdjNormal=False, CshAdjAbnormal=False, CapChg=False
...: )
Out[6]:
      AAPL US Equity
           Px_Last
2014-06-05      647.35
2014-06-06      645.57
2014-06-09       93.70
2014-06-10       94.25

```

BDH adjusted for dividends and splits:

```

In [7]: blp.bdh(
...:     'AAPL US Equity', 'Px_Last', '20140605', '20140610',
...:     CshAdjNormal=True, CshAdjAbnormal=True, CapChg=True
...: )
Out[7]:
      AAPL US Equity
           Px_Last
2014-06-05      85.45
2014-06-06      85.22
2014-06-09      86.58
2014-06-10      87.09

```

BDS example:

```

In [8]: blp.bds('AAPL US Equity', 'DVD_Hist_All', DVD_Start_Dt='20180101', DVD_End_Dt=
↪ '20180531')
Out[8]:
      declared_date      ex_date record_date payable_date  dividend_amount_
↪ dividend_frequency dividend_type
AAPL US Equity    2018-05-01  2018-05-11  2018-05-14  2018-05-17      0.73  ↪
↪      Quarter  Regular Cash

```

(continues on next page)



(continued from previous page)

AAPL US Equity	2018-02-01	2018-02-09	2018-02-12	2018-02-15	0.63	↵
↵ Quarter Regular Cash						

Intraday bars BDIB example:

```
In [9]: blp.bdib(ticker='BHP AU Equity', dt='2018-10-17').tail()
Out[9]:
```

		BHP AU Equity					
		open	high	low	close	volume	num_trds
2018-10-17	15:56:00+11:00	33.62	33.65	33.62	33.64	16660	126
2018-10-17	15:57:00+11:00	33.65	33.65	33.63	33.64	13875	156
2018-10-17	15:58:00+11:00	33.64	33.65	33.62	33.63	16244	159
2018-10-17	15:59:00+11:00	33.63	33.63	33.61	33.62	16507	167
2018-10-17	16:10:00+11:00	33.66	33.66	33.66	33.66	1115523	216

Above example works because 1) AU in equity ticker is mapped to EquityAustralia in markets/assets.yml, and 2) EquityAustralia is defined in markets/exch.yml. To add new mappings, define BBG\_ROOT in sys path and add assets.yml and exch.yml under BBG\_ROOT/markets.

New in 0.6.6 - if exchange is defined in /xbbg/markets/exch.yml, can use ref to look for relevant exchange market hours. Both ref='ES1 Index' and ref='CME' work for this example:

```
In [10]: blp.bdib(ticker='ESM0 Index', dt='2020-03-20', ref='ES1 Index').tail()
out[10]:
```

		ESM0 Index						
		open	high	low	close	volume	num_trds	↵
↵ value								
2020-03-20	16:55:00-04:00	2,260.75	2,262.25	2,260.50	2,262.00	412	157	931,
↵ 767.00								
2020-03-20	16:56:00-04:00	2,262.25	2,267.00	2,261.50	2,266.75	812	209	1,838,
↵ 823.50								
2020-03-20	16:57:00-04:00	2,266.75	2,270.00	2,264.50	2,269.00	1136	340	2,576,
↵ 590.25								
2020-03-20	16:58:00-04:00	2,269.25	2,269.50	2,261.25	2,265.75	1077	408	2,439,
↵ 276.00								
2020-03-20	16:59:00-04:00	2,265.25	2,272.00	2,265.00	2,266.50	1271	378	2,882,
↵ 978.25								

Intraday bars within market session:

```
In [11]: blp.bdib(ticker='7974 JT Equity', dt='2018-10-17', session='am_open_30').
↵tail()
Out[11]:
```

		7974 JT Equity					
		open	high	low	close	volume	num_trds
2018-10-17	09:27:00+09:00	39,970.00	40,020.00	39,970.00	39,990.00	10800	44
2018-10-17	09:28:00+09:00	39,990.00	40,020.00	39,980.00	39,980.00	6300	33
2018-10-17	09:29:00+09:00	39,970.00	40,000.00	39,960.00	39,970.00	3300	21
2018-10-17	09:30:00+09:00	39,960.00	40,010.00	39,950.00	40,000.00	3100	19
2018-10-17	09:31:00+09:00	39,990.00	40,000.00	39,980.00	39,990.00	2000	15

Corporate earnings:

```
In [12]: blp.earning('AMD US Equity', by='Geo', Eqty_Fund_Year=2017, Number_Of_
↵Periods=1)
Out[12]:
```

level	fy2017	fy2017_pct
-------	--------	------------

(continues on next page)

(continued from previous page)

Asia-Pacific	1.00	3,540.00	66.43
China	2.00	1,747.00	49.35
Japan	2.00	1,242.00	35.08
Singapore	2.00	551.00	15.56
United States	1.00	1,364.00	25.60
Europe	1.00	263.00	4.94
Other Countries	1.00	162.00	3.04

**Dividends:**

```
In [13]: blp.dividend(['C US Equity', 'MS US Equity'], start_date='2018-01-01', end_
↳date='2018-05-01')
Out[13]:
```

	dec_date	ex_date	rec_date	pay_date	dvd_amt	dvd_freq	
↳dvd_type							
C US Equity	2018-01-18	2018-02-02	2018-02-05	2018-02-23	0.32	Quarter	↳
↳Regular Cash							
MS US Equity	2018-04-18	2018-04-27	2018-04-30	2018-05-15	0.25	Quarter	↳
↳Regular Cash							
MS US Equity	2018-01-18	2018-01-30	2018-01-31	2018-02-15	0.25	Quarter	↳
↳Regular Cash							

*New in 0.1.17* - Dividend adjustment can be simplified to one parameter adjust:

- BDH without adjustment for dividends and splits:

```
In [14]: blp.bdh('AAPL US Equity', 'Px_Last', '20140606', '20140609', adjust='-')
Out[14]:
```

	AAPL US Equity
	Px_Last
2014-06-06	645.57
2014-06-09	93.70

- BDH adjusted for dividends and splits:

```
In [15]: blp.bdh('AAPL US Equity', 'Px_Last', '20140606', '20140609', adjust='all')
Out[15]:
```

	AAPL US Equity
	Px_Last
2014-06-06	85.22
2014-06-09	86.58

## 5.2 Data Storage

If `BBG_ROOT` is provided in `os.environ`, data can be saved locally. By default, local storage is preferred than Bloomberg for all queries.

Noted that local data usage must be compliant with Bloomberg Datafeed Addendum (full description in [DAPI<GO>](#)):

To access Bloomberg data via the API (and use that data in Microsoft Excel), your company must sign the ‘Datafeed Addendum’ to the Bloomberg Agreement. This legally binding contract describes the terms and conditions of your use of the data and information available via the API (the “Data”). The most fundamental requirement regarding your use of Data is that it cannot leave the local PC you use to access the BLOOMBERG PROFESSIONAL service.

Docs	
Build	
Coverage	
Quality	
License	