10. The `fxcmpy_data_reader`

This section is about the `fxcmpy_data_reader` classes

- `fxcmpy_tick_data_reader` for historical tick data and
- `fxcmpy_candles_data_reader` for historicial candles data.

10.1. The `fxcmpy_tick_data_reader`

In [1]:

```
import importlib
import datetime as dt
from fxcmpy import fxcmpy_tick_data_reader as tdr
```

The `fxcmpy_tick_data_reader` reads tick data from the FXCM servers for a given symbol and a given time window.

10.1.1. Available Symbols

There is tick data for the following symbols (currency pairs) available.

In [2]:

```
print(tdr.get_available_symbols())
```

```
('AUDCAD', 'AUDCHF', 'AUDJPY', 'AUDNZD', 'CADCHF', 'EURAUD', 'EURCHF', 'EURGBP', 'EURJPY', 'EURUSD', 'GBPCHF', '
```

10.1.2. Reading Tick Data

First, start and end dates need to be specified.

In [3]:

```
start = dt.datetime(2018, 2, 1)
end = dt.datetime(2018, 2, 15)
```

Note, that the tick data is stored in weekly packages.

In [4]:

```
dr = tdr('EURUSD', start, end, verbosity=True)
```

```
Fetching data from: https://tickdata.fxcorporate.com/EURUSD/2018/5.csv.gz
Fetching data from: https://tickdata.fxcorporate.com/EURUSD/2018/6.csv.gz
Fetching data from: https://tickdata.fxcorporate.com/EURUSD/2018/7.csv.gz
```

The resulting object is of type `fxcmpy_tick_data_reader` and has the methods `get_data()` and `get_raw_data()`.

In [5]:

```
type(dr)
```

Out[5]:

```
fxcmpy.fxcmpy_data_reader.fxcmpy_tick_data_reader
```

In [6]:

```
raw_data= dr.get_raw_data()
```

`get_raw_data()` returns the data in a `DataFrame` object with a string representation of the dates as index.

In [7]:

```
raw_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 5643477 entries, 01/28/2018 22:00:46.425 to 02/16/2018 21:59:08.423
Data columns (total 2 columns):
Bid    float64
Ask    float64
dtypes: float64(2)
memory usage: 129.2+ MB
```

`get_data()` returns the same `DataFrame` object but with a `DatetimeIndex` (note that the conversion is time consuming for large data sets).

In [8]:

```
dr.get_data().info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 5643477 entries, 2018-01-28 22:00:46.425000 to 2018-02-16 21:59:08.423000
Data columns (total 2 columns):
Bid    float64
Ask    float64
dtypes: float64(2)
memory usage: 129.2 MB
```

In [9]:

```
dr.get_data().head(10)
```

Out[9]:

	Bid	Ask
2018-01-28 22:00:46.425	1.24239	1.24282
2018-01-28 22:00:46.438	1.24236	1.24282
2018-01-28 22:00:59.117	1.24236	1.24267
2018-01-28 22:00:59.134	1.24229	1.24267
2018-01-28 22:01:25.668	1.24228	1.24266
2018-01-28 22:01:29.157	1.24259	1.24266
2018-01-28 22:01:30.164	1.24258	1.24266
2018-01-28 22:01:33.137	1.24243	1.24266
2018-01-28 22:01:33.153	1.24228	1.24266
2018-01-28 22:02:04.496	1.24228	1.24261

10.2. The `fxcmpy_candles_data _reader`

The `fxcmpy_candles_data_reader` returns candles from the FXCM servers for a given symbol. The basic usage is as before.

In [10]:

```
from fxcmpy import fxcmpy_candles_data_reader as cdr
```

In [11]:

```
print(cdr.get_available_symbols())
```

```
('AUDCAD', 'AUDCHF', 'AUDJPY', 'AUDNZD', 'CADCHF', 'EURAUD', 'EURCHF', 'EURGBP', 'EURJPY', 'EURUSD', 'GBPCHF', '
```

In [12]:

```
start = dt.datetime(2018, 2, 1)
end = dt.datetime(2018, 3, 1)
```

`period` defines the granularity of the data and must be one of `m1`, `h1` or `D1`.

In [13]:

```
period = 'm1'
```

In [14]:

```
candles = cdr('EURGBP', start, end, period, verbosity=True)
```

```
Fetching data from: https://candledata.fxcorporate.com/m1/EURGBP/2018/5.csv.gz
Fetching data from: https://candledata.fxcorporate.com/m1/EURGBP/2018/6.csv.gz
Fetching data from: https://candledata.fxcorporate.com/m1/EURGBP/2018/7.csv.gz
Fetching data from: https://candledata.fxcorporate.com/m1/EURGBP/2018/8.csv.gz
Fetching data from: https://candledata.fxcorporate.com/m1/EURGBP/2018/9.csv.gz
```

The resulting object is of type `fxcmpy_candles_data_reader` and has, similar to `fxcmpy_tick_data_reader` the methods `get_data()` and `get_raw_data()`.

In [15]:

```
type(candles)
```

Out[15]:

```
fxcmpy.fxcmpy_data_reader.fxcmpy_candles_data_reader
```

In [16]:

```
raw_data = candles.get_raw_data()
```

In [17]:

```
raw_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 35932 entries, 01/28/2018 22:00:00.000 to 03/02/2018 21:59:00.000
Data columns (total 8 columns):
BidOpen     35932 non-null float64
BidHigh     35932 non-null float64
BidLow      35932 non-null float64
BidClose    35932 non-null float64
AskOpen     35932 non-null float64
AskHigh     35932 non-null float64
AskLow      35932 non-null float64
AskClose    35932 non-null float64
dtypes: float64(8)
memory usage: 2.5+ MB
```

In [18]:

```
raw_data.head()
```

Out[18]:

DateTime	BidOpen	BidHigh	BidLow	BidClose	AskOpen	AskHigh	AskLow	AskClose
01/28/2018 22:00:00.000	0.87849	0.87861	0.87849	0.87858	0.87938	0.87938	0.87912	0.87912
01/28/2018 22:01:00.000	0.87858	0.87858	0.87833	0.87833	0.87912	0.87930	0.87912	0.87923
01/28/2018 22:03:00.000	0.87833	0.87845	0.87833	0.87842	0.87923	0.87930	0.87919	0.87929
01/28/2018 22:04:00.000	0.87842	0.87843	0.87842	0.87842	0.87929	0.87929	0.87929	0.87929
01/28/2018 22:05:00.000	0.87842	0.87877	0.87842	0.87858	0.87929	0.87942	0.87913	0.87913

In [19]:

```
data = candles.get_data()
```

In [20]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 35932 entries, 2018-01-28 22:00:00 to 2018-03-02 21:59:00
Data columns (total 8 columns):
BidOpen     35932 non-null float64
BidHigh     35932 non-null float64
BidLow      35932 non-null float64
BidClose    35932 non-null float64
AskOpen     35932 non-null float64
AskHigh     35932 non-null float64
AskLow      35932 non-null float64
AskClose    35932 non-null float64
dtypes: float64(8)
memory usage: 2.5 MB
```

`D1` for period only works for time windows before the current year

In [22]:

```
period = 'D1'
candles = cdr('EURGBP', start, end, period)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-22-ec17e517d548> in <module>()
      1 period = 'D1'
----> 2 candles = cdr('EURGBP', start, end, period)

/notebooks/pyalgo/schwed/www/fxcmpy-docu/fxcmpy/fxcmpy_data_reader.py in __init__(self, symbol, start, end, peri
    166         self.codec = 'utf-8'
    167         self.url = 'https://candledata.fxcorporate.com/%s/%s/%s/%s.csv.gz'
--> 168         self.__fetch_data__()
    169
    170

/notebooks/pyalgo/schwed/www/fxcmpy-docu/fxcmpy/fxcmpy_data_reader.py in __fetch_data__(self)
    190             if stop >= dt.datetime.now().year:
    191                 msg = "Candles with period 'D1' are restricted to years before %s"
--> 192                 raise ValueError(msg % dt.datetime.now().year )
    193             for year in range(start, stop+1):
    194                 url = 'https://candledata.fxcorporate.com/%s/%s/%s.csv.gz'

ValueError: Candles with period 'D1' are restricted to years before 2018
```

However, they can be used for years including 2017.

In [23]:

```
start = dt.datetime(2017, 7, 1)
end = dt.datetime(2017, 11, 1)
```

In [24]:

```
period = 'D1'
candles = cdr('EURGBP', start, end, period)
```

In [25]:

```
candles.get_data().info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 309 entries, 2017-01-02 22:00:00 to 2017-12-31 22:00:00
Data columns (total 8 columns):
BidOpen     309 non-null float64
BidHigh     309 non-null float64
BidLow      309 non-null float64
BidClose    309 non-null float64
AskOpen     309 non-null float64
AskHigh     309 non-null float64
AskLow      309 non-null float64
AskClose    309 non-null float64
dtypes: float64(8)
memory usage: 21.7 KB
```