



## Experiment 6

**Aim:** Analyse Financial Balance Sheet, Cash flows and Profit & Loss using FXCM Dataset.

**Theory: FXCM-** provides a wealth of information to individuals interested in the study of market fundamentals and technical. Economic calendars, news feeds, charting data and proprietary indicators enable traders to view the forex within a current or historical context. Opportunity is often found within the framework of intensive market study, and FXCM offers a robust research suite capable of satisfying even the most inquisitive currency trader or investor.

**FXCMPY Python Wrapper:** FXCM provides a RESTful API to interact with its trading platform. Among others, it allows the retrieval of historical data as well as of streaming data. In addition, it allows to place different types of orders and to read out account information. The overall goal is to allow the implementation automated, algorithmic trading programs. fxcmpy.py is a Python wrapper package for that API.

To get started with the the API and the package, a demo account with FXCM is sufficient. Account opening link:

[https://tradingstation2.fxcm.com/oauth/authenticate?client\\_id=Titan&response\\_type=code&redirect\\_uri=https://tradingstation.fxcm.com&code\\_challenge=II77w4-bRdIY7YK8tnxfgPNy0BJ2hrEt8AFSubmhqtA&code\\_challenge\\_method=S256&state=2moidm8owd](https://tradingstation2.fxcm.com/oauth/authenticate?client_id=Titan&response_type=code&redirect_uri=https://tradingstation.fxcm.com&code_challenge=II77w4-bRdIY7YK8tnxfgPNy0BJ2hrEt8AFSubmhqtA&code_challenge_method=S256&state=2moidm8owd)

In an interactive context, use a variable called TOKEN to reference to personal unique API token.

### Technical Analytics

In the current digital marketplaces of the world, the majority of short-term trading decisions are made according to the tenants of technical analysis. The study of past and present price action as a predictor of future market behaviour is an extremely popular discipline among active traders.

For those practitioners of technical trading methodologies, FXCM provides a variety of assets to aid in the study and analysis of price action:

**Charting:** Forex Charts is an application enabling traders to pull a price chart for a desired instrument, with customisable time period and indicator overlay.



A.Y.: 2022-23

Class/Sem: T.Y.B.Tech/ Sem-VI

Sub: Computational Finance

**Market Scanner:** FXCM's Market Scanner is a collection of leading indicators applied to the top forex and CFD products.

**Market Data Signals:** FXCM Market Data Signals offers traders the latest trends in volume, price and market sentiment.

In addition to the above listed resources, FXCM Plus affords traders an abundance of tools for performance enhancement. A software suite including the proprietary supplements Trading Signals, Technical Analyser and Trading Analytics gives clients a means of augmenting efficiency within the marketplace.

# Importing Data from FXCM

## Connecting to the API

```
import pandas as pd
```

```
import fxcmpy
```

```
token = "INSERT YOUR PERSONAL TOKEN HERE!"
```

```
api = fxcmpy.fxcmpy(access_token= token, log_level= "error")
```

```
api
```

```
api.get_instruments()
```

```
api.close()
```

## Currency / FX incl. Bid/Ask Spreads

```
api = fxcmpy.fxcmpy(access_token= token, log_level= "error")
```

```
api.get_instruments()
```

```
api.get_candles('EUR/USD')
```

```
api.get_candles('USD/EUR')
```

```
api.get_candles('GBP/USD')
```



A.Y.: 2022-23  
api.close()

Class/Sem: T.Y.B.Tech/ Sem-VI

Sub: Computational Finance

## Setting the Frequency (Intraday)

```
api = fxcmpy.fxcmpy(access_token= token, log_level= "error")
```

```
api.get_candles('EUR/USD', period = "D1")
```

```
api.get_candles('EUR/USD', period = "m1")
```

```
api.close()
```

## Setting the Time Period

```
api = fxcmpy.fxcmpy(access_token= token, log_level= "error")
```

```
api.get_candles('EUR/USD', period = "D1", number = 10000)
```

```
api.get_candles('EUR/USD', period = "D1", start = "2001-01-01")
```

```
api.get_candles('EUR/USD', period = "H1", number = 10000)
```

```
api.get_candles('EUR/USD', period = "H1", start = "2017-05-01", end = "2018-12-31")
```

```
api.get_candles('EUR/USD', period = "m1", number = 10000)
```

```
api.get_candles('EUR/USD', period = "m1", start = "2019-08-05", end = "2019-08-20")
```

```
api.get_candles('EUR/USD', period = "D1", start = "2001-01-01", end = "2018-12-31",  
columns = ["bidclose", "askclose"])
```

```
api.close()
```

## Stock Indexes incl. Bid/Ask Spreads

```
api = fxcmpy.fxcmpy(access_token= token, log_level= "error")
```

```
api.get_instruments()
```

```
api.get_candles('SPX500', period = "D1", number = 10000)
```

```
api.get_candles('US30', period = "D1", number = 10000)
```

```
api.get_candles('GER30', period = "H1", start = "2019-08-15", end = "2019-08-20")
```

```
api.close()
```

## Commodities incl. Bid/Ask Spreads



A.Y.: 2022-23

Class/Sem: T.Y.B.Tech/ Sem-VI

Sub: Computational Finance

```
api = fxcmpy.fxcmpy(access_token= token, log_level= "error")
```

```
api.get_instruments()
```

```
api.get_candles('USOil', period = "D1", number = 10000)
```

```
api.get_candles('XAU/USD', period = "D1", number = 10000)
```

```
api.get_candles('XAG/USD', period = "D1", number = 10000)
```

```
api.get_candles('CORNF', period = "H1", number = 10000)
```

```
api.close()
```

```
## Cryptocurrencies incl. Bid/Ask Spreads
```

```
api = fxcmpy.fxcmpy(access_token= token, log_level= "error")
```

```
api.get_instruments()
```

```
api.get_candles("BTC/USD", period = "D1", start = "2002-01-01", end = "2018-12-31")
```

```
api.get_candles("ETH/USD", period = "m1", number = 10000)
```

```
api.close()
```

```
## Streaming real-time Data (Part 1)
```

```
api = fxcmpy.fxcmpy(access_token= token, log_level= "error")
```

```
api.subscribe_market_data("EUR/USD")
```

```
api.get_subscribed_symbols()
```

```
api.get_last_price("EUR/USD")
```

```
api.get_prices("EUR/USD")
```

```
import time
```

```
while True:
```

```
    time.sleep(1)
```

```
    print(api.get_last_price("EUR/USD").name, api.get_last_price("EUR/USD").Ask)
```

```
api.unsubscribe_market_data('EUR/USD')
```

```
api.get_subscribed_symbols()
```



A.Y.: 2022-23  
api.close()

Class/Sem: T.Y.B.Tech/ Sem-VI

Sub: Computational Finance

## Streaming real-time Data (Part 2)

```
api = fxcmpy.fxcmpy(access_token= token, log_level= "error")
```

```
def print_data(data, dataframe):
```

```
    print('%3d | %s | %s, %s, %s, %s, %s'
```

```
          % (len(dataframe), data['Symbol'],
```

```
            pd.to_datetime(int(data['Updated']), unit='ms'),
```

```
            data['Rates'][0], data['Rates'][1], data['Rates'][2],
```

```
            data['Rates'][3]))
```

```
api.subscribe_market_data('EUR/USD', (print_data,))
```

```
api.unsubscribe_market_data('EUR/USD')
```

```
api.close()
```

**Lab assignments to be done by students:**

- 1. Import Data from FXCM**
- 2. Connect to the REST API of FXCM financial data provider**
- 3. Get Currency, Foreign exchange including Bid and Ask Spreads**
- 4. Set the Frequency setting (Intraday)**
- 5. Set the Time Period specific setting**
- 6. Get Stock Indexes including Bid and Ask Spreads**
- 7. Get commodities incl. Bid and Ask Spreads**
- 8. Get Cryptocurrencies including Bid and Ask Spreads**
- 9. Stream real-time Data by subscribing to market data**
- 10. Stream real-time market Data in precise format**

# Importing Data from FXCM

## Connecting to the API

In [1]:

```
#pip install fxcmpy
```

In [2]:

```
import pandas as pd
import fxcmpy
```

```
-----
ModuleNotFoundError                                Traceback (most recent call last)
<ipython-input-2-53c9ab1f0de2> in <cell line: 2>()
      1 import pandas as pd
----> 2 import fxcmpy
```

ModuleNotFoundError: No module named 'fxcmpy'

NOTE: If your import is failing due to a missing package, you can manually install dependencies using either !pip or !apt.

To view examples of installing some common dependencies, click the "Open Examples" button below.

In [ ]:

```
token = "INSERT YOUR PERSONAL TOKEN HERE!"
```

In [ ]:

```
api = fxcmpy.fxcmpy(access_token= token, log_level= "error")
```

In [ ]:

```
api
```

In [ ]:

```
api.get_instruments()
```

In [ ]:

```
api.close()
```

In [ ]:

## Currency / FX incl. Bid/Ask Spreads

In [ ]:

```
api = fxcmpy.fxcmpy(access_token= token, log_level= "error")
```

```
api = fxcmpy.fxcmpy(access_token= token, log_level= "error",
```

```
In [ ]:
```

```
api.get_instruments()
```

```
In [ ]:
```

```
api.get_candles('EUR/USD')
```

```
In [ ]:
```

```
api.get_candles('USD/EUR')
```

```
In [ ]:
```

```
api.get_candles('GBP/USD')
```

```
In [ ]:
```

```
api.close()
```

```
In [ ]:
```

## Setting the Frequency (Intraday)

```
In [ ]:
```

```
api = fxcmpy.fxcmpy(access_token= token, log_level= "error")
```

```
In [ ]:
```

```
api.get_candles('EUR/USD', period = "D1")
```

```
In [ ]:
```

```
api.get_candles('EUR/USD', period = "m1")
```

```
In [ ]:
```

```
api.close()
```

```
In [ ]:
```

## Setting the Time Period

```
In [ ]:
```

```
api = fxcmpy.fxcmpy(access_token= token, log_level= "error")
```

```
In [ ]:
```

```
api.get_candles('EUR/USD', period = "D1", number = 10000)
```

```
In [ ]:
```

```
api.get_candles('EUR/USD', period = "D1", start = "2001-01-01")
```

```
In [ ]:
```

```
api.get_candles('EUR/USD', period = "H1", number = 10000)
```

```
In [ ]:
```

```
api.get_candles('EUR/USD', period = "H1", start = "2017-05-01", end = "2018-12-31")
```

```
In [ ]:
```

```
api.get_candles('EUR/USD', period = "m1", number = 10000)
```

```
In [ ]:
```

```
api.get_candles('EUR/USD', period = "m1", start = "2019-08-05", end = "2019-08-20")
```

```
In [ ]:
```

```
api.get_candles('EUR/USD', period = "D1", start = "2001-01-01", end = "2018-12-31", columns = ["bidclose", "askclose"])
```

```
In [ ]:
```

```
api.close()
```

```
In [ ]:
```

## Stock Indexes incl. Bid/Ask Spreads

```
In [ ]:
```

```
api = fxcmpy.fxcmpy(access_token= token, log_level= "error")
```

```
In [ ]:
```

```
api.get_instruments()
```

```
In [ ]:
```

```
api.get_candles('SPX500', period = "D1", number = 10000)
```

```
In [ ]:
```

```
api.get_candles('US30', period = "D1", number = 10000)
```

```
In [ ]:
```

```
api.get_candles('GER30', period = "H1", start = "2019-08-15", end = "2019-08-20")
```

```
In [ ]:
```

```
api.close()
```

```
In [ ]:
```

## Commodities incl. Bid/Ask Spreads

```
In [ ]:
```

```
api = fxcmpy.fxcmpy(access_token= token, log_level= "error")
```

```
In [ ]:
```

```
api.get_instruments()
```

```
In [ ]:
```



```
api.get_candles('USOil', period = "D1", number = 10000)
```

```
In [ ]:
```

```
api.get_candles('XAU/USD', period = "D1", number = 10000)
```

```
In [ ]:
```

```
api.get_candles('XAG/USD', period = "D1", number = 10000)
```

```
In [ ]:
```

```
api.get_candles('CORNF', period = "H1", number = 10000)
```

```
In [ ]:
```

```
api.close()
```

```
In [ ]:
```

## Cryptocurrencies incl. Bid/Ask Spreads

```
In [ ]:
```

```
api = fxcmpy.fxcmpy(access_token= token, log_level= "error")
```

```
In [ ]:
```

```
api.get_instruments()
```

```
In [ ]:
```

```
api.get_candles("BTC/USD", period = "D1", start = "2002-01-01", end = "2018-12-31")
```

```
In [ ]:
```

```
api.get_candles("ETH/USD", period = "m1", number = 10000)
```

```
In [ ]:
```

```
api.close()
```

```
In [ ]:
```

## Streaming real-time Data (Part 1)

```
In [ ]:
```

```
api = fxcmpy.fxcmpy(access_token= token, log_level= "error")
```

```
In [ ]:
```

```
api.subscribe_market_data("EUR/USD")
```

```
In [ ]:
```

```
api.get_subscribed_symbols()
```

```
In [ ]:
```

```
api.get_last_price("EUR/USD")
```

```
In [ ]:
```

```
api.get_prices("EUR/USD")
```

```
In [ ]:
```

```
import time
```

```
In [ ]:
```

```
while True:
    time.sleep(1)
    print(api.get_last_price("EUR/USD").name, api.get_last_price("EUR/USD").Ask)
```

```
In [ ]:
```

```
api.unsubscribe_market_data('EUR/USD')
```

```
In [ ]:
```

```
api.get_subscribed_symbols()
```

```
In [ ]:
```

```
api.close()
```

```
In [ ]:
```

## Streaming real-time Data (Part 2)

```
In [ ]:
```

```
api = fxcmpy.fxcmpy(access_token= token, log_level= "error")
```

```
In [ ]:
```

```
def print_data(data, dataframe):
    print('%3d | %s | %s, %s, %s, %s'
          % (len(dataframe), data['Symbol'],
             pd.to_datetime(int(data['Updated']), unit='ms'),
             data['Rates'][0], data['Rates'][1], data['Rates'][2],
             data['Rates'][3]))
```

```
In [ ]:
```

```
api.subscribe_market_data('EUR/USD', (print_data,))
```

```
In [ ]:
```

```
api.unsubscribe_market_data('EUR/USD')
```

```
In [ ]:
```

```
api.close()
```

```
In [ ]:
```