8. The `fxcmpy_order` Class

This section introduces the `fxcmpy_order` class which is created by different operations.

In [1]:

```
import fxcmpy
import pandas as pd
import datetime as dt
con = fxcmpy.fxcmpy(config_file='fxcm.cfg')
```

The methods `fxcmpy.create_entry_order()`, `fxcmpy.create_market_sell_order()`, `fxcmpy.create_market_buy_order()` and `fxcmpy.open_trade()` return an instance of the `fxcmpy_order` class. Existing orders are stored in the attribute `orders`, canceled or executed orders are stored in the attribute `old_orders`, both as instance of the `fxcmpy_order` class.

8.1. Order Ids

In [2]:

```
order = con.create_entry_order(symbol='USD/JPY', is_buy=True,
                               amount=300, limit=112,
                               is_in_pips = False,
                               time_in_force='GTC', rate=110,
                               stop=None, trailing_step=None)
```

The `ids` of existing orders are returned by `con.get_order_ids()`:

In [3]:

```
order_ids = con.get_order_ids()
print(order_ids)
```

```
[404809369]
```

The last order id:

In [4]:

```
order_id = order_ids[-1]
```

`fxcmpy.orders` is a `dict` object with the order `ids` as keys and the order object as value.

In [5]:

```
con.orders
```

Out[5]:

```
{404809369: <fxcmpy.fxcmpy_order.fxcmpy_order at 0x7ff2d5a52240>}
```

8.2. Order Objects

To get an existing order object, one can use `con.get_order()`.

In [6]:

```
order = con.get_order(order_id)
```

In [7]:

```
print(order)
```

```
accountId:        2815291
accountName:      02815291
amountK:          300
```

```
buy:              110
currency:         USD/JPY
currencyPoint:    27.2726
isBuy:            True
isELSOrder:       False
isEntryOrder:     True
isLimitOrder:     True
isNetQuantity:    False
isStopOrder:      False
limit:            112
limitPegBaseType: -1
limitRate:        112
ocoBulkId:        0
orderId:          404809369
range:            0
sell:             0
status:           Waiting
stop:             0
stopMove:         0
stopPegBaseType:  -1
stopRate:         0
time:             2018-06-07 13:52:30.579000
timeInForce:      GTC
type:             LE
```

8.3. Get and Set

The `fxcmpy_order` class has **get methods** for all attributes, for example:

In [8]:

```
order.get_amount()
```

Out[8]:

```
300
```

In [9]:

```
order.get_isBuy()
```

Out[9]:

```
True
```

In [10]:

```
order.get_sell()
```

Out[10]:

```
0
```

Moreover, the class has **set methods** for the following attributes:

- `amount`: `set_amount()`
- `limit`: `set_limit_rate()`
- `range`: `set_range()`
- `rate`: `set_rate()`
- `stop`: `set_stop_rate()`
- `trailing_step`: `set_trailing_step()`

In [11]:

```
order.set_amount(600)
```

In [13]:

```
order.set_stop_rate(108, is_in_pips=False)
```

In [14]:

```
con.get_orders().T
```

Out[14]:

	0
accountId	2815291
accountName	02815291
amountK	600
buy	110
currency	USD/JPY
currencyPoint	54.5303
expireDate	
isBuy	True
isELSOrder	False
isEntryOrder	True
isLimitOrder	True
isNetQuantity	False
isStopOrder	False
limit	0
limitPegBaseType	-1
limitRate	112
ocoBulkId	0
orderId	404809369
range	0
ratePrecision	3
sell	0
status	1
stop	108
stopMove	0
stopPegBaseType	-1
stopRate	108
t	3
time	06072018135425810
timeInForce	GTC
tradeId	178168067
type	LE

8.4. Status and Deletion

A `fxcmpy_order` object has its own `delete` method.

In [15]:

```
order.get_status()
```

Out[15]:

```
'Waiting'
```

In [16]:

```
order.delete()
```

In [17]:

```
order.get_status()
```

Out[17]:

```
'Canceled'
```

8.5. Canceled Orders

The **canceled order** can still be found in the `old_orders` attribute.

In [18]:

```
con.old_orders
```

Out[18]:

```
{404809369: <fxcmpy.fxcmpy_order.fxcmpy_order at 0x7ff2d5a52240>,
 404809502: <fxcmpy.fxcmpy_order.fxcmpy_order at 0x7ff2d5a52ac8>}
```

In [19]:

```
con.close()
```