# ig-markets-api-python-library Documentation

*Release 0.0.6*

**FemtoTrader**

**Apr 29, 2018**

# Contents


**1 IG Markets API - Python Library** **3**
- 1.1 REST API . . . 3
- 1.2 STREAM API . . . 3
- 1.3 More . . . 3
- 1.4 Install . . . 4
- 1.5 Work in progress . . . 4
- 1.6 Thanks to . . . 4
- 1.7 Other related projects . . . 5
- 1.8 See also . . . 5

**2 Indices and tables** **9**

CHAPTER 1

# IG Markets API - Python Library

A lightweight Python library that can be used to connect to the IG Markets REST and STREAM API with a LIVE or DEMO account.

IG Markets provide Retail Spread Betting and CFD accounts for trading Equities, Forex, Commodities, Indices and much more.

## 1.1 REST API

You can use the IG Markets HTTP / REST API to submit trade orders, open positions, close positions and view market sentiment.

## 1.2 STREAM API

You can use the IG Markets STREAM API to get realtime price, balance...

## 1.3 More

Full details about the API along with information about how to open an account with IG can be found at the link below:

http://labs.ig.com/

## 1.4 Install

### 1.4.1 From Python package index

```
$ pip install trading_ig
```

### 1.4.2 From source

Get latest version using Git

```
$ git clone https://github.com/ig-python/ig-markets-api-python-library
$ cd ig-markets-api-python-library
$ python setup.py install https://github.com/ig-python/ig-markets-api-python-library
```

or

```
$ pip install git+https://github.com/ig-python/ig-markets-api-python-library
```

## 1.5 Work in progress

This project is not a IG Markets project. Use it at your own risk.

There is still some room for improvement, fix issue

see :

- http://labs.ig.com/node/98
- https://labs.ig.com/node/28
- http://www.andlil.com/forum/script-api-ig-stream-rest-t10091-10.html
- https://github.com/ig-python/ig-markets-api-python-library/issues

## 1.6 Thanks to

- Lewis Barber
- ixta
- Chris
- colombao
- gianluca.finocchiaro
- Weswit
- Dimitri John Ledkov
- RRMXkun
- redbullpeter
- David Sabater Dinter

- Oscar Neira
- Wez Pyke
- and all other contributors...

## 1.7 Other related projects

- igtrade (and forks)
  - https://github.com/maroxe/igtrade
  - https://github.com/falex69/igtrade
  - https://github.com/yopibou/L3

## 1.8 See also

### 1.8.1 IG Markets REST API - Python Library

You can use the IG Markets HTTP / REST API to submit trade orders, open positions, close positions and view market sentiment.

Full details about the API along with information about how to open an account with IG can be found at the link below:

http://labs.ig.com/

#### How To Use The Library

Using this library to connect to the IG Markets API is extremely easy. All you need to do is import the IGService class, create an instance, and call the methods you wish to use. There is a method for each endpoint exposed by their API. The code sample below shows you how to connect to the API, switch to a secondary account and retrieve all open positions for the active account.

**Note:** The secure session with IG is established when you create an instance of the IGService class.

```
from trading_ig import IGService
from trading_ig.config import config

ig_service = IGService(config.username, config.password, config.api_key, config.acc_
↪type)
ig_service.create_session()

account_info = ig_service.switch_account(config.acc_number, False) # not necessary
print(account_info)

open_positions = ig_service.fetch_open_positions()
print("open_positions:\n%s" % open_positions)

print("")

epic = 'CS.D.EURUSD.MINI.IP'
resolution = 'D'
num_points = 10
```

```
response = ig_service.fetch_historical_prices_by_epic_and_num_points(epic, resolution,
↪ num_points)
df_ask = response['prices']['ask']
print("ask prices:\n%s" % df_ask)
```

with `trading_ig_config.py`

```
class config(object):
    username = "YOUR_USERNAME"
    password = "YOUR_PASSWORD"
    api_key = "YOUR_API_KEY"
    acc_type = "DEMO" # LIVE / DEMO
    acc_number = "ABC123"
```

Config can also be set as environment variable

```
export IG_SERVICE_USERNAME="..."
export IG_SERVICE_PASSWORD="..."
export IG_SERVICE_API_KEY="..."
export IG_SERVICE_ACC_TYPE="DEMO" # LIVE / DEMO
export IG_SERVICE_ACC_NUMBER="..."
```

it should display:

```
open_positions:
Empty DataFrame
Columns: []
Index: []

ask prices:
                        Open     High      Low    Close
DateTime
2014:11:18-00:00:00  1.24510  1.25465  1.24442  1.25330
2014:11:19-00:00:00  1.25332  1.26013  1.25127  1.25461
2014:11:20-00:00:00  1.25463  1.25760  1.25048  1.25427
2014:11:21-00:00:00  1.25428  1.25689  1.23755  1.23924
2014:11:23-00:00:00  1.23640  1.23770  1.23607  1.23725
2014:11:24-00:00:00  1.23864  1.24453  1.23830  1.24390
2014:11:25-00:00:00  1.24389  1.24877  1.24026  1.24743
2014:11:26-00:00:00  1.24744  1.25322  1.24443  1.25077
2014:11:27-00:00:00  1.25078  1.25244  1.24569  1.24599
2014:11:28-00:00:00  1.24598  1.24909  1.24269  1.24505
```

Many IGService methods return Python Pandas DataFrame, Series or Panel.

### Cache queries requests-cache

Set CachedSession using:

```
from datetime import datetime, timedelta
import requests_cache
session = requests_cache.CachedSession(cache_name='cache', backend='sqlite', expire_
↪after=timedelta(hours=1))
# set expire_after=None if you don't want cache expiration
# set expire_after=0 if you don't want to cache queries
```

CachedSession can be applied globally on IGService

```
ig_service = IGService(config.username, config.password, config.api_key, config.acc_
↪type, session)
ig_service.create_session()
```

or just for a given method (like fetching prices)

```
epic = 'CS.D.EURUSD.MINI.IP'
resolution = 'D'
start_date = '2014-12-15'
end_date = '2014-12-20'
response = ig_service.fetch_historical_prices_by_epic_and_date_range(epic, resolution,
↪ start_date, end_date, session)
```

### 1.8.2 IG Markets STREAM API - Python Library

You can use the IG Markets STREAM API to get realtime price, balance...

Full details about the API along with information about how to open an account with IG can be found at the link below:

http://labs.ig.com/

#### How To Use The Library

Run sample using:

```
$ python sample/stream_ig.py
INFO:requests.packages.urllib3.connectionpool:Starting new HTTPS connection (1): demo-
↪api.ig.com
INFO:ig_stream:Starting connection with https://demo-apd.marketdatasystems.com
L1:CS.D.USDJPY.CFD.IP: Time 20:35:43 - Bid 119.870 - Ask 119.885
L1:CS.D.GBPUSD.CFD.IP: Time 20:35:46 - Bid 1.51270 - Ask 1.51290
----------HIT CR TO UNSUBSCRIBE AND DISCONNECT FROM     LIGHTSTREAMER-----------
L1:CS.D.USDJPY.CFD.IP: Time 20:35:43 - Bid 119.870 - Ask 119.885
L1:CS.D.USDJPY.CFD.IP: Time 20:35:48 - Bid 119.871 - Ask 119.886
L1:CS.D.GBPUSD.CFD.IP: Time 20:35:48 - Bid 1.51271 - Ask 1.51291
L1:CS.D.USDJPY.CFD.IP: Time 20:35:48 - Bid 119.870 - Ask 119.885
L1:CS.D.GBPUSD.CFD.IP: Time 20:35:49 - Bid 1.51270 - Ask 1.51290

INFO:lightstreamer:Unsubscribed successfully
WARNING:lightstreamer:Server error
DISCONNECTED FROM LIGHTSTREAMER
```

### 1.8.3 ToDo

- config file (YAML or .py which could be overwritten by environment variable)

see

http://opensourcehacker.com/2012/12/13/configuring-your-python-application-using-environment-variables/

CHAPTER 2

# Indices and tables

- genindex
- modindex
- search