Blockchain programmation

ESILV 2018/2019

Ordre du jour

APIs
~5mn

Crypto currency exchanges
~5mn

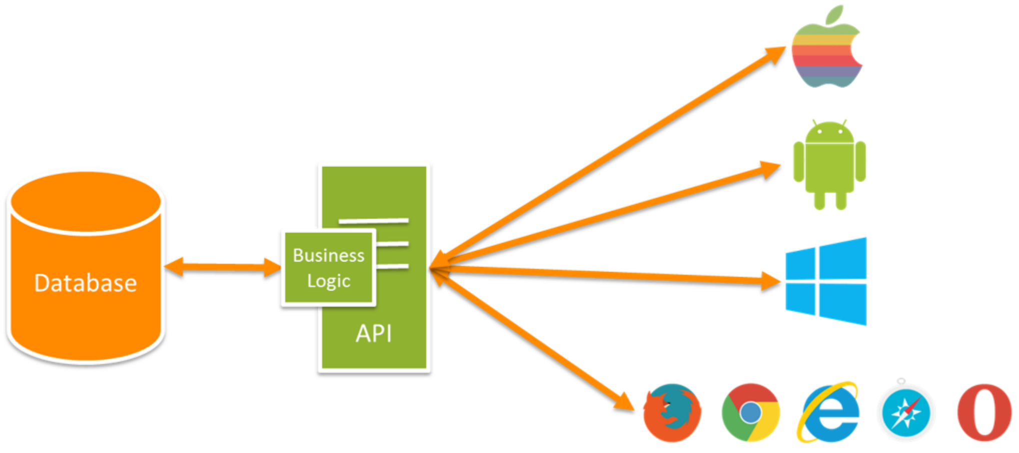
Common trading strategies
~10mn

Using crypto exchange API
~1h15mn

APIs

Application Programmable Interfaces

- Traditionnal web design is backend <-> Frontend <-> Browser
- Very powerful for human centric web
- Hard to automate interactions between machines
- API make it easier to automate processes between different web ressources



Application Programmable Interfaces

- SOAP: Simple Object Access Protocol. Very stricly defined protocol, based on XML
- Websocket: Mostly read only data streams, faster than other methods. Much less implemented
- RESTful: Representational State Transfer. Based on HTTP, less strictly defined, uses JSON a lot
- Methods:
  - GET: Retrieve resources
  - POST: Send resources
  - PUT: Modify resources
  - DELETE: Delete resources
  - Custom methods…

Crypto currency exchanges

POLONIEX

PAYMIUM
MONEY OVER IP

GDAX

Crypto exchanges

- There are hundreds of crypto exchanges
- When programming robots, points of attention:
  - Is the exchange safe? (risk of loss of funds)
  - Is the volume worth it?
  - What cryptos are traded on it?
  - How stable is the API?
  - How well documented is the API?

BINANCE

BITSTAMP

kraken

Common indicators

- Bid: The most someone is willing to pay for an asset
- Ask: The least somebody is willing to receive for an asset
- Candles: Bundles of transactions
  - Duration: Length in time of the candle
  - Open: First transaction in the candle
  - Close: Last transaction
  - High/Low: extreme price values of the candle

Common trading strategies

Swing trading

- Trying to predict future movements of the market
- Buying/Selling depending on expected outcomes
- Useful tools:
  - Data collection APIs
  - Analytics
  - Order book management

Arbitrage

- Trying to gain from market inefficiency
- EG: Buying BTC at 1000 euros on exchange A, selling at the same time for 1005 euros on exchange B
- Requires various API libraries
- Requires pools of funds spread around
- Real time is key

Using crypto exchanges APIs

Using APIs

Prerequisites
- Use Python 3
- No precompiled module, write the REST calls yourself
- Use Binance or Coinbase
- Create a function for each task
- Do not store your credentials on your github!

Tasks list - GET
- Create a git repository and share it with the teacher
- Get a list of all available cryptocurrencies and display it
- Create a function to display the ’ask’ or ‘bid’ price of an asset. Direction and asset name as parameters
  def getDepth(direction='ask', pair = 'BTCUSD')
- Get order book for an asset

Using APIs

Tasks list – GET

- Create a function to read agregated trading data (candles)
  def refreshDataCandle(pair = 'BTCUSD', duration = '5m')
- Create a sqlite table to store said data (schema attached in the next slide)
- Store candle data in the db
- Modify function to update when new candle data is available
- Create a function to extract all available trade data
  def refreshData(pair = 'BTCUSD')
- Store the data in sqlite

Tasks list – POST

- Create an order
  def createOrder(api_key, secret_key, direction, price, amount, pair = 'BTCUSD_d', orderType = 'LimitOrder')
- Cancel an order
  def cancelOrder(api_key, secret_key, uuid)

Sqlite schema

Keeping track of updates:

```
CREATE TABLE last_checks(Id INTEGER PRIMARY
KEY, exchange TEXT, trading_pair TEXT, duration
TEXT, table_name TEXT, last_check INT, startdate INT,
last_id INT);
```

Data candles:

```
setTableName = str(exchangeName + "_" + pair + "_"
+ duration)
tableCreationStatement = """CREATE TABLE """ +
setTableName + """(Id INTEGER PRIMARY KEY, date
INT, high REAL, low REAL, open REAL, close REAL,
volume REAL, quotevolume REAL, weightedaverage
REAL, sma_7 REAL, ema_7 REAL, sma_30 REAL,
ema_30 REAL, sma_200 REAL, ema_200 REAL)"""
```

Full data set:

```
setTableName = str(exchangeName + "_" + pair)
tableCreationStatement = """CREATE TABLE """ +
setTableName + """(Id INTEGER PRIMARY KEY, uuid
TEXT, traded_btc REAL, price REAL, created_at_int
INT, side TEXT)""”
```

References

Wikipedia page for APIs
https://fr.wikipedia.org/wiki/Interface_de_programmation

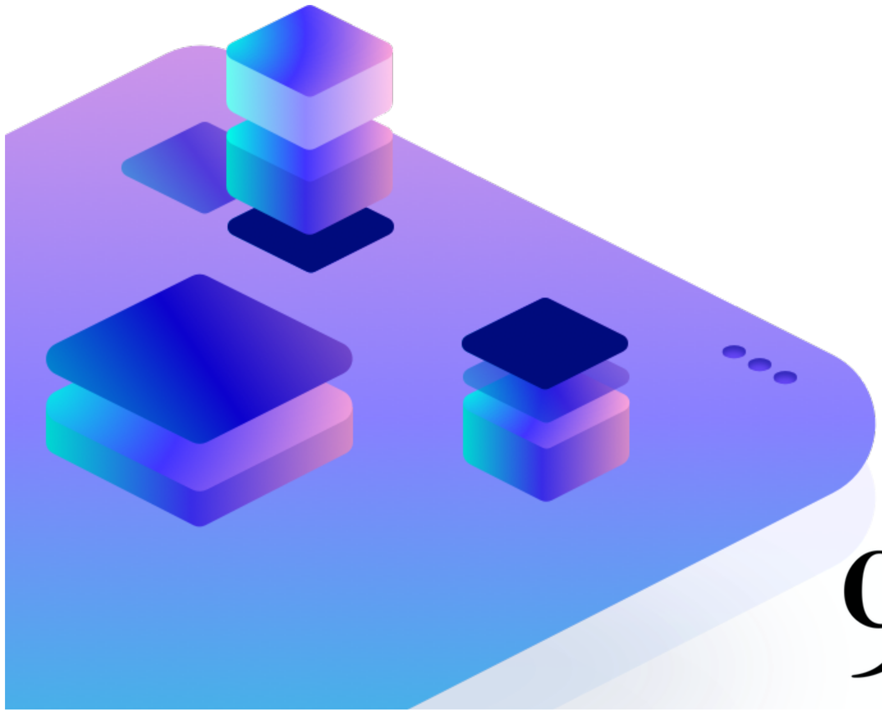
Using requests in Python
https://www.pythonforbeginners.com/requests/using-requests-in-python

Binance API Documentation
https://github.com/binance-exchange/binance-official-api-docs/blob/master/rest-api.md

Coinbase pro API documentation
https://docs.pro.coinbase.com/

Thank you

For your attention !



klsn.io

97

Twitter: @97network
Hello@97.network
Station F, 5 parvis Alan Turing, 75013 Paris
Github.com/97network