

Cryptobörse: Wie kann ich in Python den Ticker der Push API (WAMP) von Poloniex abonnieren?

geschrieben von Andreas Potthoff | 27. März 2017

Da ich gerade eine Trader-Applikation für Poloniex in Python schreibe, wollte ich dazu als erstes den Ticker der Push API abonnieren. Ich habe dazu das Modul Autobahn in Python installiert und den folgenden Code verwendet. Hier wird der gesamte Ticker-Feed über das WAMP-Protokoll des Websockets abonniert und die empfangenen Daten (Events) werden ausgegeben. Das klappt prima!

Getestet mit: Python 3.5.2 @ Ubuntu 16.04 LTS

Code:

```
from autobahn.asyncio.wamp import ApplicationSession
from autobahn.asyncio.wamp import ApplicationRunner
from asyncio import coroutine

class PoloniexComponent(ApplicationSession):
    def onConnect(self):
        self.join(self.config.realm)

    @coroutine
    def onJoin(self, details):
        def onTicker(*args):
            print("Ticker event received:", args)

        try:
            yield from self.subscribe(onTicker, 'ticker')
        except Exception as e:
            print("Could not subscribe to topic:", e)
```

```
def main():
    runner = ApplicationRunner("wss://api.poloniex.com:443",
    "realm1")
    runner.run(PoloniexComponent)
```

```
if __name__ == "__main__":
    main()
```

Ausgabe:

```
Ticker event received: ('ETH_ETC', '0.14829502', '0.14992249',
'0.14829510',      '-0.04851949',      '12292.09070919',
'80914.07777792', 0, '0.15614805', '0.14798000')
```

```
Ticker event received: ('USDT_ETC', '1.69796894',
'1.69796456', '1.68289334', '-0.00495273', '50045.81062071',
'29268.11748844', 0, '1.74392238', '1.68000000')
```

```
Ticker event received: ('BTC_CLAM', '0.00103996',
'0.00103996', '0.00102894', '0.02329059', '6.67321322',
'6532.51214055', 0, '0.00106603', '0.00099600')
```

```
Ticker event received: ('BTC_XCN', '0.00000257', '0.00000257',
'0.00000251', '-0.04104477', '8.59036050', '3316114.77166474',
0, '0.00000280', '0.00000250')
```

```
Ticker event received: ('USDT_LTC', '3.64300000',
'3.66651211', '3.64300000', '0.00192519', '2495.39248941',
'688.99598487', 0, '3.67999050', '3.59000000')
```

```
Ticker event received: ('USDT_NXT', '0.02430000',
'0.02629471', '0.02400000', '-0.08963942', '3555.89759389',
'136298.49703595', 0, '0.02774462', '0.02430000')
```

```
Ticker event received: ('XMR_HYP', '0.00009300', '0.00010314',
'0.00009000', '-0.15462230', '23.89571026', '241095.35982673',
0, '0.00011499', '0.00009300')
```

```
Ticker event received: ('XMR_NXT', '0.00732377', '0.00779997',
'0.00719002', '-0.35367360', '143.71435777', '15777.31974579',
0, '0.01133138', '0.00732377')
```

```
Ticker event received: ('BTC_ETH', '0.01923009', '0.01923250',
'0.01923009',      '0.00972015',      '3762.76051615',
'195213.35234572', 0, '0.01950000', '0.01902000')
```

```
Ticker event received: ('BTC_DAO', '0.00019103', '0.00019306',
'0.00019107', '-0.00639758', '37.37047925', '194598.39669886',
0, '0.00019492', '0.00018900')
```

```
Ticker event received: ('ETH_STEEM', '0.12000000',
```

```

'0.12771693', '0.11820535', '-0.05064845', '1131.24971868',
'9017.54958625', 0, '0.13468721', '0.12000000')
Ticker event received: ('BTC_ETC', '0.00286215', '0.00286206',
'0.00286203', '-0.02824461', '2723.26141525',
'927741.49933883', 0, '0.00300899', '0.00285000')
Ticker event received: ('ETH_ETC', '0.14829502', '0.14992247',
'0.14829511', '-0.04851949', '12292.09070919',
'80914.07777792', 0, '0.15614805', '0.14798000')
Ticker event received: ('USDT_ETC', '1.69796894',
'1.69796453', '1.68289334', '-0.00495273', '50045.81062071',
'29268.11748844', 0, '1.74392238', '1.68000000')
Ticker event received: ('BTC_XMR', '0.00554600', '0.00556799',
'0.00555000', '0.24788213', '6990.80381868',
'1412686.14372726', 0, '0.00557071', '0.00427288')
Ticker event received: ('USDT_LTC', '3.64300000',
'3.66651208', '3.64300000', '0.00192519', '2495.39248941',
'688.99598487', 0, '3.67999050', '3.59000000')
Ticker event received: ('BTC_ETH', '0.01923009', '0.01923226',
'0.01923009', '0.00972015', '3762.76051615',
'195213.35234572', 0, '0.01950000', '0.01902000')
Ticker event received: ('BTC_FCT', '0.00429347', '0.00431739',
'0.00429346', '0.03462842', '1408.22052281',
'340507.08137862', 0, '0.00440000', '0.00381938')
Ticker event received: ('BTC_VOX', '0.00007348', '0.00007419',
'0.00007341', '-0.00284977', '171.75237449',
'2366663.48717554', 0, '0.00007749', '0.00006606')
Ticker event received: ('BTC_DAO', '0.00019103', '0.00019306',
'0.00019108', '-0.00639758', '37.37047925', '194598.39669886',
0, '0.00019492', '0.00018900')
Ticker event received: ('ETH_LSK', '0.02715264', '0.02715264',
'0.02609935', '0.00565333', '1535.89611701', '56262.64389120',
0, '0.02800000', '0.02587547')
Ticker event received: ('ETH_ETC', '0.14829502', '0.14992245',
'0.14829511', '-0.04851949', '12292.09070919',
'80914.07777792', 0, '0.15614805', '0.14798000')
Ticker event received: ('USDT_ETC', '1.69796894',
'1.69796451', '1.68289334', '-0.00495273', '50045.81062071',
'29268.11748844', 0, '1.74392238', '1.68000000')

```

Die entsprechenden Labels für diese Daten sind in der Reihenfolge:

currencyPair, last, lowestAsk, highestBid, percentChange,
baseVolume, quoteVolume, isFrozen, 24hrHigh, 24hrLow

Jetzt dürfte es kein Problem mehr sein diese Daten mit Python weiterzuverarbeiten.