Cointrader Documentation

Release 0.5.0

Torsten Irländer

Nov 08, 2018

Contents

Contents:

Cointrader

Cointraider development has stopped! If you are interested in continuing the development please write me an email.

Cointrader is a CLI trading application for crypto currencies on the Poloniex exchange written in the Python programming language. Cointrader can be used for semiautomatic guided trading.

It is released under the MIT License and Copyright 2017 by Torsten Irländer. The license text ist distributed in full text in the LICENSE file that should have come with your installation or git clone.

Danger: You could loose money. Use cointrader at your own risk! I do not accept any responsibility for any losses.

Cointrader is Alpha quality! It is not meant to be used for serious trading yet. It is still in an early development phase and has a bunch full of known defects. Expect all aspects of the application to change in the future. Cointrader is not roundly tested.

If you want to help cointrader to become the best Free Software trading bot your contribution is highly appreciated! Find details on how to participate in the documentation.

- Status: **Alpha**
- Free software: MIT license
- Source: https://github.com/toirl/cointrader
- Documentation: https://cointrader.readthedocs.io

If you like the program, I am looking forward to a donation :)

- DASH: XypsuUMRTioV7bHSVzSDkNgihtr1gfiqAr
- BTC : 1L5xtVirGVpDL7958SPaHb6p9dHZoaQ2Cz

1.1 Features

- Automatic trading. Cointrader will buy and sell following emitted trading signals.
- Semiautomatic trading. Cointrader just emits trading signals. You finally decide if you want to follow the signals.
- Paper trading. Just simlate trading. Do not actually place real orders.
- Trade logbook
- Profit/Loss analysis (Bot vs. Market)
- Backtesting. Check how good your strategy performs on historic charts.
- Explore exchanges and find interesting markets to trade on.
- Show your balances.

1.2 Planned

- Implement a working trading strategy!
- Support more exchanges
- Pluggable external trading strategies
- Risk- and Money management, Stop loss limits, Take profit limits

1.3 Motivation

This program exists because I want to learn more about automatic trading based on a technical analysis of charts. I am no expert on trading or crypto currencies! I am a professional Python programmer who stuck his nose into the crypto coin and trading world in 2017 and who was directly fascinated on this topic. After reading some books on technical analysis I decided to write this program to learn more about how automatic trading works.

1.4 Credits

This package was created with Cookiecutter and the audreyr/cookiecutter-pypackage project template.

CHAPTER 2

Installation

2.1 Stable release

To install Cointrader, run this command in your terminal:

```
$ pip install cointrader
```

This is the preferred method to install Cointrader, as it will always install the most recent stable release.

If you don't have pip installed, this Python installation guide can guide you through the process.

2.2 From sources

The sources for Cointrader can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/toirl/cointrader
```

Or download the tarball:

```
$ curl  -OL https://github.com/toirl/cointrader/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

2.3 Configuration

Contraider uses a configuration file to set some important settings like the key and the secret for the API on the configured exchange. A configuration looks like this and must be located in your *$HOME* directory with the filename

.cointrader.ini. Please make sure you have created a configuration file.:

```
[DEFAULT]
# Set default exchange which will be used for trading.
# Currently onyl Poloniex is supported!
exchange = poloniex

[poloniex]
# See https://poloniex.com/apiKeys for more details.
api_key = YOUR-API-KEY-HERE
api_secret = YOUR-API-SECRET-HERE


#
## Default logging configuration of the application.
#
[loggers]
keys = root, cointrader

[handlers]
keys = console

[formatters]
keys = generic

[logger_root]
level = ERROR
handlers = console

[logger_cointrader]
level = INFO
handlers =
qualname = cointrader

[handler_console]
class = StreamHandler
args = (sys.stderr,)
level = NOTSET
formatter = generic

[formatter_generic]
format = %(asctime)s %(levelname)-5.5s [%(name)s][%(threadName)s] %(message)s
```

Usage

The cointrader application is CLI application and provides some basic commands which are hopefully useful for your daily trading activity.

To get general help to the application please use the help flag:

```
cointrader --help
```

3.1 Trading

You can start trading by using the following command. Cointrader expects a valid currency pair as argument and the start amount of BTC to the start command. Please not that the naming of the currency pair is depended on the configured exchange.:

```
cointrader start BTC_DASH
```

Cointrader will try to load a previously created bot for the given market from the database and build the current state of the bot from the trade log. If no bot can be loaded a new bot will be created.

You can restrict the amount of coins and BTC the bot will use on trading by setting the *–btc* or *–coins* parameter. This will set the the initial amount of BTC and coins the bot will use for trading. **Tip:** You can use the *Exchange* to calculate the amount of BTC from given dollar.

> **Warning:** On default cointrader will use use the complete available amount of BTC and coins for trading for the new bot! Make sure you restrict the amount of coins and BTC the bot will use for trading!

With no further options cointrader will work on a chart with a resolution of 30min. The resolution can be changed by using the *–resolution* option.

You can set a timeframe to define the trading time of cointrader. A start and end of the timeframe can be set by using the *–start* and *–end* option which takes a datetime argument in the format "YYYY-MM-DD HH:MM:SS". On default cointrader will start to trade immediately and will trade until you stop cointrader.

Cointrader use different trading strategies. You can choose which strategy to use by setting the *–strategy* option. On default cointrader will use a very defensive "Wait-Strategy" which will only emit "Wait" signals. So no buys or sells a replaced.

Note: Cointrader currently has no working trading strategy. It is the current objectiv of the defvelopment to come up with a working profitable strategy.

Trading can be done in backtest mode. To start trading in backtest mode set the *–backtest* flag. See *Backtesting* for more details.

3.1.1 Paper Trade

A paper trade is a simulated trade. Cointrader can simlate buying and selling coins without actual involving real money. This is great to test you trading strategy without any risk. To start cointrader in paper trade mode you must set the *–papertrade* flag.

3.1.2 Interactive trading

Without any further arguments cointrader will start an *interactive trading session*. Cointrader will emit trading signals (BUY, WAIT, SELL) and waits for your decision what to do.

3.1.3 Automatic trading

If you want to start your trading session in a automatic session you can set the *–automatic* flag. Cointrader will then automatically take action on the emitted trading signals. In automatic mode the resolution will determine between two trading actions.

3.2 Balance

Cointrader can show your current balance at the configured exchange by invoking the balance command:

```
contrader balance
```

This will give you an output like this:

```
CUR          total     btc_value
-------------------------------
DASH:  14.34446293    0.34583237
BTC :   0.04910656    0.04910656
-------------------------------
TOTAL BTC:            0.39493893
TOTAL USD:          465.392085719
```

3.3 Explore

Cointrader can explore the different markets on the given exchange and will result the most interesting markets to trade on for the last 24H:

```
contrader explore
```

On default cointrader will look for the top three volume and profit markets and only lists those markets which are in the top three in both categories. The command will give you an output like this:

```
BTC_DASH      4.47%       2190.7 https://poloniex.com/exchange#btc_dash
BTC_ETH       3.21%       5138.0 https://poloniex.com/exchange#btc_eth
```

If the command gives no output means that there are no markets in the top three which met bot criteria. In this situation you can either use the *–top* attribute to increase the amount of markets which are considered as interesting.

Alternatively you can use the *–order-by-volume* and *–order-by-profit* flag to only look on profit or volume markets.

3.4 Exchange

Exchange is a simple helper command to calculate how many BTC you get for a certain amount of USD:

```
cointrader change 50                                                                    ⌐
↪                                              2.Mär.17 23.09
-> 50.0$ ~ 0.03999086BTC # 2017-03-02
```

Backtesting

Backtesting is the act of testing a strategy on real chart data from the past. This can be used to measure the performance of your trading strategy in comparison to the market.

Doing backtests with cointrader is easy by setting the *–backtest* flag:

```
cointrader start \
        --backtest \
        --strategy Trend \
        --resolution 2h \
        --automatic \
        --coins 10 \
        --start "2017-03-18 00:00:00" \
        --end "2017-03-21 19:00:00" \
        BTC_DASH
```

Despite of this single flag the bot will work as usual. It will trade on the given market for the defined timeframe and resolution. After the backtest has finished a small statistic on this trading run is shown:

```
2017-03-21 22:18:39,411 INFO  [cointrader.bot][MainThread] Creating new bot BTC_DASH
2017-03-21 22:18:40,405 INFO  [cointrader.bot][MainThread] 2017-03-18 00:00:00: INIT␣
→0.0 BTC 10.0 COINS
2017-03-21 22:18:40,604 INFO  [cointrader.bot][MainThread] 2017-03-21 00:00:00: SELL␣
→10.0 @ 0.09662999 earned -> 0.966058325025 BTC
2017-03-21 22:18:40,662 INFO  [cointrader.bot][MainThread] Backtest finished
```

At the end the tradelog will be displayed to see the trading activity of the bot:

```
Tradelog:
+---------------------+------+------------+-------+--------+-----+--------------+
| DATE                | TYPE | RATE       | COINS | COINS' | BTC | BTC'         |
+---------------------+------+------------+-------+--------+-----+--------------+
| 2017-03-18 00:00:00 | INIT | 0.0881     | 10.0  | --     | 0.0 | --           |
| 2017-03-21 00:00:00 | SELL | 0.09662999 | 10.0  | --     | --  | 0.966058325025 |
+---------------------+------+------------+-------+--------+-----+--------------+
```

And finally a statistic of the performance of the strategy is shown:

```
Statistic:
+-----------+-------------------+-------------------+----------+
|           | 2017-03-18 00:00:00 | 2017-03-21 18:00:00 | CHANGE % |
+-----------+-------------------+-------------------+----------+
| COINTRADER | 0.881             | 0.966058325025    | 8.8047   |
| MARKET    | 0.881             | 0.8361            | -5.3702  |
+-----------+-------------------+-------------------+----------+
```

The statistic compares the performance of the bot with the market evaluation. This is done by comparing BTC values. The value is defined as:

```
value = amount of coins * rate + btc
```

The rate refers to the current rate of the market at the begin and the end of the timeframe.

This makes it easy to see if the bot makes more or less money for you.

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

5.1 Types of Contributions

5.1.1 Report Bugs

Report bugs at https://github.com/toirl/cointrader/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

5.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" and "help wanted" is open to whoever wants to implement it.

5.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "enhancement" and "help wanted" is open to whoever wants to implement it.

5.1.4 Write Documentation

Cointrader could always use more documentation, whether as part of the official Cointrader docs, in docstrings, or even on the web in blog posts, articles, and such.

5.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/toirl/cointrader/issues.

If you are proposing a feature:

- Explain in detail how it would work.

- Keep the scope as narrow as possible, to make it easier to implement.

- Remember that this is a volunteer-driven project, and that contributions are welcome :)

5.2 Get Started!

Ready to contribute? Here's how to set up *cointrader* for local development.

1. Fork the *cointrader* repo on GitHub.

2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/cointrader.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv cointrader
$ cd cointrader/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 cointrader tests
$ python setup.py test or py.test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

5.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.

2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.

3. The pull request should work for Python 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/toirl/cointrader/pull_requests and make sure that the tests pass for all supported Python versions.

5.4 Tips

To run a subset of tests:

```
$ py.test tests.test_cointrader
```

Credits

6.1 Development Lead

- Torsten Irländer <torsten.irlaender@googlemail.com>

6.2 Contributors

None yet. Why not be the first?

History

7.1 0.5.0 (not yet released)

Overall code refectoring. Moved indicators into its own modul. Cleanup strategy code and reworked the interctive mode.

New Features:

• Added option to detach a bot into automatic mode and to reatach a bot into intercativ mode.

Other:

• **Make bot more robust against wrong user input**

    – Check given market name

    – Check given resolution

7.2 0.4.0 (2017-03-16)

First version with real trading functionality. However **cointrader has no working trading strategy yet**. So the default strategy will do nothing than waiting :). However you can use the interactive mode to buy and sell coins if you want to.

• Added automatic trading. Cointrader will follow the emitted signals from the strategy.

• Added Papertrading. Do trading without any risk. Cointrader will simulate trading.

• Added Tradelog.

7.3 0.3.1 (2017-03-11)

Bugfix release.

- Fixed issue #4 (https://github.com/toirl/cointrader/issues/4) - Use absolute imports. - Renamed contrader modul into bot to prevent namespace issues. - Added missing requirement of the 'requests' package.

7.4 0.3.0 (2017-03-02)

- Added backtest functionality. Cointrader can simulate trading in backtest mode. In this mode the trade is done on historic chart data. This is useful to check the performance of your trading strategy. Please note that the backtest is not 100% accurate as buy and sell prices are based on the closing price of the currency. This is idealistic and will not reflect the whole market situation.

- Added new "exchange" command. Can be used to calculate how many BTC you get for a certain amount of USD.

7.5 0.2.0 (2017-02-26)

- Improved "Usage" documentation
- Changed format of configuration file from JSON to standard python configuration file (.ini)
- Added "balance" command
- Added "explore" command

7.6 0.1.0 (2017-02-21)

- First release on PyPI

CHAPTER 8

Indices and tables

- genindex
- modindex
- search

Index

B

Backtest, 8

I

interactive trading, 8

P

Papertrading, 8

S

Strategy, 7