
quandl*fund_xlsx* Documentation

Release 0.4.0

Robert Rennison

Apr 21, 2020

Contents

1	quandl_fund_xlsx	3
1.1	Features	3
1.2	Installation	4
1.3	Configuration	4
1.4	Usage of the quandl_fund_xlsx CLI command	5
1.5	Local Development	5
1.6	How to get help contribute or provide feedback	6
1.7	Credits	6
2	Installation	7
2.1	Stable release	7
2.2	From sources	7
3	Contributing	9
3.1	Types of Contributions	9
3.2	Get Started!	10
3.3	Pull Request Guidelines	11
3.4	Tips	11
4	Credits	13
4.1	Development Lead	13
4.2	Contributors	13
5	History	15
5.1	0.1.1 (2017-08-31)	15
5.2	0.1.2 (2017-08-31)	15
5.3	0.1.3 (2017-08-31)	15
5.4	0.1.4 (2017-11-06)	15
5.5	0.1.6 (2018-01-26)	15
5.6	0.1.7 (2018-05-10)	16
5.7	0.1.8 (2018-05-24)	16
5.8	0.1.9 (2018-06-11)	16
5.9	0.1.10 (2018-10-29)	16
5.10	0.2.0 (2018-11-13)	16
5.11	0.2.1 (2018-11-13)	16
5.12	0.2.2 (2018-11-13)	17
5.13	0.2.3 (2018-12-29)	17

5.14	0.3.0 (2019-09-12)	17
5.15	0.3.1 (2019-11-11)	17
5.16	0.3.2 (2020-03-31)	17
5.17	0.4.0 (2020-04-21)	17

6 Indices and tables **19**

Contents:

A unofficial CLI tool which uses the Quandl API and the Sharadar Essential Fundamentals Database to extract financial fundamentals, Sharadar provided ratios as well as calculate additional ratios. Results are written to an Excel Workbook with a separate worksheet per ticker analysed.

- Free software: Apache Software License 2.0
- Documentation: https://quandl_fund_xlsx.readthedocs.io.

1.1 Features

For a given ticker, fundamental data is obtained using the Quandl API and the Sharadar Fundamentals database. This data is then used to calculate various useful, financial ratios. The ratios include

- Profitability indicators
- Financial leverage indicators
- Free and Operating Cash flow indicators.

Some REIT specific ratios such as FFO are very roughly approximated. These specific ratios are only roughly approximated since certain data, namely Real estate sales data for the period does not appear to be available via the API (It's often buried in the footnotes of these companies filings).

The output excel worksheet for each ticker processed is divided into three main areas:

- Sharadar statement indicators. This is data obtained from the three main financial statements; the Income Statement, the Balance Sheet and the Cash Flow Statement.
- Sharadar Metrics and Ratio Indicators. These are quandl provided financial ratios.

- Calculated Metrics and Ratios. These are calculated by the package from the Sharadar data provided and tabulated by the statement indicators and the 'Metrics and Ratio' indicators.

The python Quandl API provides the ability to return data within python pandas dataframes. This makes calculating various ratios as simple as dividing two variables by each other.

The calculations support the data offered by the free sample database (formerly referred to by Sharadar as the SF0 database), and the paid for SF1 database. The coverage universe is the same for both the sample data and the paid database. The key difference being, support as well as a much richer set of so-called Dimensions (timeperiods). For example the sample data is taken from the annual filings of companies, whereas the paid data allows for Trailing Twelve Month as well as quarterly data.

Note: For quarterly data, many of the ratios using income and cash flow statement values in the numerator will be inaccurate when using quarterly data e.g EBITDA/Interest expense or Total Debt/ Cash Flow from Operations.

docs/Tech-1.png

Fig. 1: The generated Excel workbook with one sheet per ticker.

docs/Tech-2.png

Fig. 2: Some bespoke metrics and ratios calculated based on Sharadar fundamentals.

1.2 Installation

```
pip install quandl_fund_xlsx
```

1.3 Configuration

You will need a Quandl API key. This may be obtained by signing up, for free at [Quandl Signup](#). The key will then be available under "profile" when logging into Quandl. This key allows for access to sample data for many of the datasets.

If you have a key for the free sample data set the QUANDL_API_SF0_KEY environment variable to the value of your key.

If you have paid for access to the Sharadar fundamentals data set, then set the QUANDL_API_SF1_KEY in the environment.

```
export QUANDL_API_SF0_KEY='YourQuandlAPIKey'  
  
or  
  
export QUANDL_API_SF1_KEY='YourQuandlAPIKey'
```

For windows the setx command is used to set environment variables..

1.4 Usage of the quandl_fund_xlsx CLI command

```
quandl_fund_xlsx -h
quandl_fund_xlsx

Usage:
quandl_fund_xlsx (-i <ticker-file> | -t <ticker>) [-o <output-file>]
                                                    [-y <years>] [-d
↪<sharadar-db>]
                                                    [--dimension <dimension>]

quandl_fund_xlsx.py (-h | --help)
quandl_fund_xlsx.py --version

Options:
-h --help                Show this screen.
-i --input <file>       File containing one ticker per line
-t --ticker <ticker>    Ticker symbol
-o --output <file>      Output file [default: stocks.xlsx]
-y --years <years>     How many years of results (max 7 with SF0) [default: 5]
-d --database <database> Sharadar Fundamentals database to use, SF0 or
                                                                    SF1 [default: SF0]
--dimension <dimension> Sharadar database dimension, ARY, MRY, ART, MRT [default:
↪MRY]
--version                Show version.
```

```
quandl_fund_xlsx -t INTC -o intc-MRY.xlsx
{'--database': 'SF0',
 '--input': None,
 '--output': 'INTC-MRY.xlsx',
 '--ticker': 'INTC',
 '--years': '5'}
('Ticker =', 'INTC')
2017-08-22 06:08:59,751 INFO      Processing the stock INTC
2017-08-22 06:09:06,012 INFO      Processed the stock INTC

ls -lh excel_files
total 12K
-rw-rw-r-- 1 test test 8.7K Aug 22 06:09 intc-MRY.xlsx
```

1.5 Local Development

This section is only of relevance if you wish to hack on the code yourself, perhaps to add new ratios or display other Sharadar provided data values.

It's recommended to setup a virtual environment and perform the installation within this. Use pip to install the requirements but not the package.

```
pip install -r requirements_dev.txt

# Run the CLI by running as a module
python -m quandl_fund_xlsx.cli -t MSFT

# Run the tests
pytest
```

If you wish to install the package locally within either a virtualenv or globally this can be done once again using pip.

```
pip install -e .  
  
# Now the CLI is installed within our environment and should be on the  
# path  
quandl_fund_xlsx -t MSFT
```

1.6 How to get help contribute or provide feedback

See the *contribution submission and feedback guidelines* <ref-contributing>

1.7 Credits

This package was created with [Cookiecutter](#) and the [audreyr/cookiecutter-pypackage](#) project template.

2.1 Stable release

To install `quandl_xlsx_cli`, run this command in your terminal:

```
$ pip install quandl_fund_xlsx
```

This is the preferred method to install `quandl_xlsx_cli`, 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 `quandl_xlsx_cli` can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/robren/quandl_fund_xlsx
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/robren/quandl_fund_xlsx/tarball/master
```

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

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


Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

3.1 Types of Contributions

3.1.1 Report Bugs

Report bugs at https://github.com/robren/quandl_fund_xlsx/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.

3.1.2 Fix Bugs

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

3.1.3 Implement Features

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

3.1.4 Write Documentation

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

3.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/robren/quandl_fund_xlsx/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 :)

3.2 Get Started!

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

1. Fork the *quandl_fund_xlsx* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/quandl_fund_xlsx.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 quandl_fund_xlsx
$ cd quandl_fund_xlsx/
$ 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 quandl_fund_xlsx 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.

3.3 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/robren/quandl_fund_xlsx/pull_requests and make sure that the tests pass for all supported Python versions.

3.4 Tips

To run a subset of tests:

```
$ py.test tests.test_quandl_fund_xlsx
```


4.1 Development Lead

- Robert Rennison <rob@robren.net>

4.2 Contributors

None yet. Why not be the first?

5.1 0.1.1 (2017-08-31)

- First release on PyPI.

5.2 0.1.2 (2017-08-31)

- Change logging to INFO from DEBUG

5.3 0.1.3 (2017-08-31)

- Minor tweak to Return the correct version

5.4 0.1.4 (2017-11-06)

- Removed the `-dimension` CLI keyword. Now uses Most Recent Year (MRY) for SF0 database and Most Recent Trailing 12 Months (MRT) for the SF1 database
- Fix to avoid the Pandas future warning about decrementing `df.rename_axis` and using `df.rename`

5.5 0.1.6 (2018-01-26)

Now uses the `get_table` methods from the `quandl_api`.

5.6 0.1.7 (2018-05-10)

- Fix bug where the dataframe returned from quandl qas not being sorted
- Added EPS and EPS diluted.

5.7 0.1.8 (2018-05-24)

- Fix bug where the SF0 subscription data was not being returned.
- With the discontinuation of the Sharadar Time series API at the end of March 2018, the codes for the free fundamental subscription SF0 database changed. Subscribers to the SF0 data now use the SHARADAR/SF1 code in the get_table accesses.

5.8 0.1.9 (2018-06-11)

- Added back support for the -dimension CLI option.

5.9 0.1.10 (2018-10-29)

- Added some new Cash Flow related ratios and corrected the LTDEBT ratios
- Changed the default to be the paid SF1 Database as this is the one I'm using and testing. Requires a separate free SF0 subscription to test SFO. All of the API calls whether the user has an SFI paid membership or SF0 use the SF1 codes.

5.10 0.2.0 (2018-11-13)

- After learning that the sample data API now allows _all_ of the same indicators as those available using the paid SF! aPI key I was able to remove a lot of special case code for the Sample data KEY. The paid KEY allows for many more dimensions to be queried.
- The CLI now defaults back to using the sample data SF0 API key.
- Added a number of Cash Flow from Operations based metrics as well as some Free Cash Flow based metrics.
- Added a development test which uses the API and a sample data or SF0 API key to extract ratios for AAPL.
- Added Excess Cash Margin ratio.

5.11 0.2.1 (2018-11-13)

- Minor security fix, requests version now $\geq 2.20.0$
- Minor documentation cleanup

5.12 0.2.2 (2018-11-13)

- Add support for the MRQ and ARQ dimensions.
- Correct error in calculating CAGR when the data was given in quarterly increments.
- Correctly reference the Excel spreadsheet example figures in the README.

5.13 0.2.3 (2018-12-29)

- Check for the presence of the QUANDL_API_SF0_KEY or the QUANDL_API_SF1_KEY environment variable depending on which database the user is requesting to use.

5.14 0.3.0 (2019-09-12)

- Refactored by using and manipulating the pandas dataframe as it's returned from quandl/Sharadar. The dates are rows and the columns are the "observations" ie the revenue, income etc. The dataframe is transposed prior to writing to excel so that the data is in the typically viewed format of dates as columns and the observations as rows.

5.15 0.3.1 (2019-11-11)

- Added some metrics favored by Kenneth J Marshall, author of "Good Stocks Cheap: Value investing with confidence for a lifetime of Stock Market Outperformance"

5.16 0.3.2 (2020-03-31)

- Added the working capital value from the balance sheet Note: For finance companies and REITS this is not provided by the API.

5.17 0.4.0 (2020-04-21)

- Added a summary sheet as the first sheet of a workbook. This is conditionally formatted to highlight the best and worst values for each of the summarized metrics of each ticker in the workbook. The summary table is an excel table and thus each column can be sorted to put best on top or worst on top.

CHAPTER 6

Indices and tables

- `genindex`
- `modindex`
- `search`