
ecbxrate Documentation

Release 0.2.3

Alastair Houghton

May 12, 2017

Contents

1	What is this?	3
2	Usage	5
3	Code Documentation	7
3.1	ecbxrate package	7
4	Indices and tables	9

This document refers to version 0.2.3

CHAPTER 1

What is this?

`ecbxrate` is a Python module that fetches exchange rate data from the European Central Bank (ECB) and stores it in a local database for later querying.

CHAPTER 2

Usage

ecbxrate installs a script that you can use from the command line; e.g.:

```
$ ecbxrate -d sqlite:///tmp/ecb.sqlite3 initialise
Initialised with 127246 rates covering 4075 days
$ ecbxrate -d sqlite:///tmp/ecb.sqlite3 status
Last updated for 2014-11-27
$ ecbxrate -d sqlite:///tmp/ecb.sqlite3 2014-11-27 EUR GBP
At 2014-11-27, EUR:GBP = 0.792000
$ ecbxrate -d sqlite:///tmp/ecb.sqlite3 2014-11-26 EUR GBP 26.99
At 2014-11-26, EUR:GBP = 0.790950
EUR 26.99 = GBP 21.35
$ ecbxrate -d sqlite:///tmp/ecb.sqlite3 latest ISK GBP
At 2008-12-09, ISK:GBP = 0.00300379
```

Note in particular the last one, which is present to make the point that there may be rates in the data set that are not up to date; for instance, the ECB no longer (as of the 9th of December 2008) publishes a rate for ISK, so the last valid exchange rate between ISK and GBP is that calculated from the currencies' respective rates on that day in 2008. (ISK is a particularly interesting example because the Icelandic króna is still in use.)

The first thing you need to do when using this package is to use the script to initialise your database; this will create the necessary tables and will import data in bulk from the ECB's historic data set.

You also need to make sure that you tell *ecbxrate* to update itself at least once a day, otherwise you will need to initialise the database again. You can either update from the command line using e.g.:

```
$ ecbxrate -d sqlite:///tmp/ecb.sqlite3 update
Updated 32 rates for 2014-11-27
```

or you can use the Python API:

```
>>> import ecbxrate, datetime
>>> store = ecbxrate.ExchangeRateStore('sqlite:///tmp/ecb.sqlite3')
>>> store.update()
(32, datetime.date(2014, 11, 27))
```

Note that the ECB usually updates its reference rates daily at 3pm CET; before that time, the latest rate they will have available will be for the *previous day*. Also, the ECB does not necessarily update its rates every day; in particular, on public holidays there may be no published rate.

If you want to obtain the latest rate from code, you can write e.g.:

```
>>> store.get_rate('EUR', 'GBP', as_of_date='latest')
(datetime.date(2014, 11, 27), Decimal('0.792000'))
```

If you want the latest rate before or on a particular date, you can write e.g.:

```
>>> store.get_rate('EUR', 'GBP', as_of_date=datetime.date(2014, 11, 27),
...               closest_rate=ecbxrate.BEFORE)
(datetime.date(2014, 11, 27), Decimal('0.792000'))
```

You can also ask for the earliest rate on or after a particular date:

```
>>> store.get_rate('EUR', 'GBP', as_of_date=datetime.date(2015, 10, 10),
...               closest_rate=ecbxrate.AFTER)
(datetime.date(2015, 10, 12), Decimal('0.740100'))
```

Note that if you don't specify `closest_rate` in the above code and no rate exists on that date, `get_rate` will return `(date, None)`:

```
>>> store.get_rate('GBP', 'TRL', as_of_date=datetime.date(2014, 11, 27))
(datetime.date(2014, 11, 27), None)
>>> store.get_rate('GBP', 'TRL', as_of_date=datetime.date(2014, 11, 27),
...               closest_rate=True)
(datetime.date(2004, 12, 31), Decimal('2604354.301113'))
```

The rates returned always have at least 6dp of precision. If rounding to 6dp would mean that there are fewer than 6 significant figures, then they are rounded to 6 significant figures instead. This should be sufficiently accurate for most purposes.

CHAPTER 3

Code Documentation

Contents:

ecbxrate package

Classes

CHAPTER 4

Indices and tables

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