time-series-metadata Documentation

Maximilian Gruber, Björn Ludwig, Bang Xiang Yong, Benedikt See

Aug 31, 2022

Getting started:

1 time-series-buffer - a metrological time-series buffer 3

2 TimeSeriesBuffer - the reference 5

3 Indices and tables 7

Python Module Index 9

Index 11

time-series-buffer is a Python software package developed by software developers and researchers from Physikalisch-Technische Bundesanstalt (Germany) as part of the joint European Research Project EMPIR 17IND12 Met4FoF[1] and the German research project FAMOUS[2].

For the *time-series-buffer* homepage go to GitHub[3].

time-series-buffer is written in Python 3 and strives to run with all Python versions with upstream support[4]. Currently it is tested to work with Python 3.5 to 3.8.

Python package[5] Documentation Status[6]

[1] https://met4fof.eu
[2] https://famous-project.eu
[3] https://github.com/PTB-PSt1/time-series-buffer
[4] https://devguide.python.org/#status-of-python-branches
[5] https://pypi.org/project/time-series-buffer/
[6] https://time-series-buffer.readthedocs.io/en/latest/

CHAPTER 1

time-series-buffer - a metrological time-series buffer

This package provides support for time-series buffering based on the build-in Python `collections.deque`.

The package is developed and maintained at the "Physikalisch-Technische Bundesanstalt" by Björn Ludwig and Maximilian Gruber.

CHAPTER 2

TimeSeriesBuffer - the reference

class `time_series_buffer.buffer.`**`TimeSeriesBuffer`**(*maxlen=10*, *return_type='array'*)

Custom buffer class, that allows to save streams of time-series with uncertainty in timestamps and values. Acts like a FIFO buffer.

add(*data=None*, *time=nan*, *time_unc=0.0*, *val=nan*, *val_unc=0.0*)

Append one or more new datapoints to the buffer. A datapoint consists of the tuple (time, time_uncertainty, value, value_uncertainty).

Parameters

- **data** (*iterable of iterables with shape (N, M) (default: None)*) – If given, all other kwargs are ignored.
  - M==2 (pairs): assumed to be like (time, value)
  - M==3 (triple): assumed to be like (time, value, value_unc)
  - M==4 (4-tuple): assumed to be like (time, time_unc, value, value_unc)
- **time** (*float, or iterable of float/ufloat (default: np.nan)*) – Timestamp(s) to be added.
- **time_unc** (*float, or iterable of float (default: 0.0)*) – Uncertainty(ies) of the timestamp(s) to be added.
- **val** (*(iterable of) float/ufloat (default: np.nan)*) – Value(s) to be added.
- **val_unc** (*(iterable of) float (default: 0.0)*) – Uncertainty(ies) of the value(s) to be added.
- **time_unc, val, val_unc need to be of same shape, but uncertainties can be omitted.** (*time,*) –

pop(*n_samples=1*)

Return the next *n_samples* from the left side of the buffer.

View the latest *n* additions to the buffer. Returns the format that was specified during init of the buffer.

Parameters **n** (*int (default: 1)*) – How many datapoints to return.

Returns

Return type Depends on return_type, see `__init__()` for details

show(*n_samples=1*)

View the latest *n_samples* additions to the buffer. Returns the format that was specified during init of the buffer.

Parameters **n_samples** (*int (default: 1)*) – How many samples to return. Return all samples in buffer, if set to -1.

Returns

Return type Depends on return_type, see `__init__()` for details

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

t

time_series_buffer.buffer, 5

Index

A

`add()` (*time_series_buffer.buffer.TimeSeriesBuffer method*), 5

P

`pop()` (*time_series_buffer.buffer.TimeSeriesBuffer method*), 5

S

`show()` (*time_series_buffer.buffer.TimeSeriesBuffer method*), 6

T

`time_series_buffer.buffer` (*module*), 5

`TimeSeriesBuffer` (*class in time_series_buffer.buffer*), 5