

---

# **time-series-metadata Documentation**

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

**Aug 01, 2023**



---

Getting started:

---

<b>1</b>	<b>time-series-metadata</b>	<b>3</b>
<b>2</b>	<b>MetaData - the metrologically enabled time-series metadata scheme</b>	<b>7</b>
<b>3</b>	<b>Indices and tables</b>	<b>9</b>
	<b>Python Module Index</b>	<b>11</b>
	<b>Index</b>	<b>13</b>



time-series-metadata is a Python software package developed jointly by software developers and researchers from [Physikalisch-Technische Bundesanstalt](https://www.ifm.eng.cam.ac.uk/) (Germany) and [Institute for Manufacturing](https://met4fof.eu)<sup>1</sup> (UK) as part of the joint European Research Project [EMPIR 17IND12 Met4FoF](https://famous-project.eu)<sup>2</sup> and the German research project [FAMOUS](https://github.com/PTB-M4D/time-series-metadata)<sup>3</sup>.

For the *time-series-metadata* homepage go to [GitHub](https://github.com)<sup>4</sup>.

*time-series-metadata* is written in Python 3 and strives to run with all Python versions with upstream support<sup>5</sup>. Currently it is tested to work with Python 3.7 to 3.10.

---

<sup>1</sup> <https://www.ifm.eng.cam.ac.uk/>

<sup>2</sup> <https://met4fof.eu>

<sup>3</sup> <https://famous-project.eu>

<sup>4</sup> <https://github.com/PTB-M4D/time-series-metadata>

<sup>5</sup> <https://devguide.python.org/#status-of-python-branches>



CircleCI<sup>6</sup> Documentation Status<sup>7</sup> DOI<sup>8</sup>

## 1.1 A metrologically enabled time-series metadata scheme

*time-series-metadata* is a Python implementation of a metadata scheme for time-series with measurement uncertainties. It is developed jointly by software developers and researchers from [Physikalisch-Technische Bundesanstalt](https://www.ptb.de)<sup>9</sup> (Germany) and [Institute for Manufacturing](https://www.ifm.eng.cam.ac.uk/)<sup>10</sup> (UK) as part of the joint European Research Project [EMPIR 17IND12 Met4FoF](https://met4fof.eu)<sup>11</sup> and the German research project [FAMOUS](https://famous-project.eu)<sup>12</sup>.

*time-series-metadata* is written in Python 3 and strives to run with all Python versions with upstream support<sup>13</sup>. Currently, it is tested to work with Python 3.8 to 3.11.

## 1.2 Scheme

The following image illustrates an abstract representation of a time series:

---

<sup>6</sup> <https://circleci.com/gh/PTB-M4D/time-series-metadata>

<sup>7</sup> <https://time-series-metadata.readthedocs.io/en/latest/>

<sup>8</sup> <https://doi.org/10.5281/zenodo.3935859>

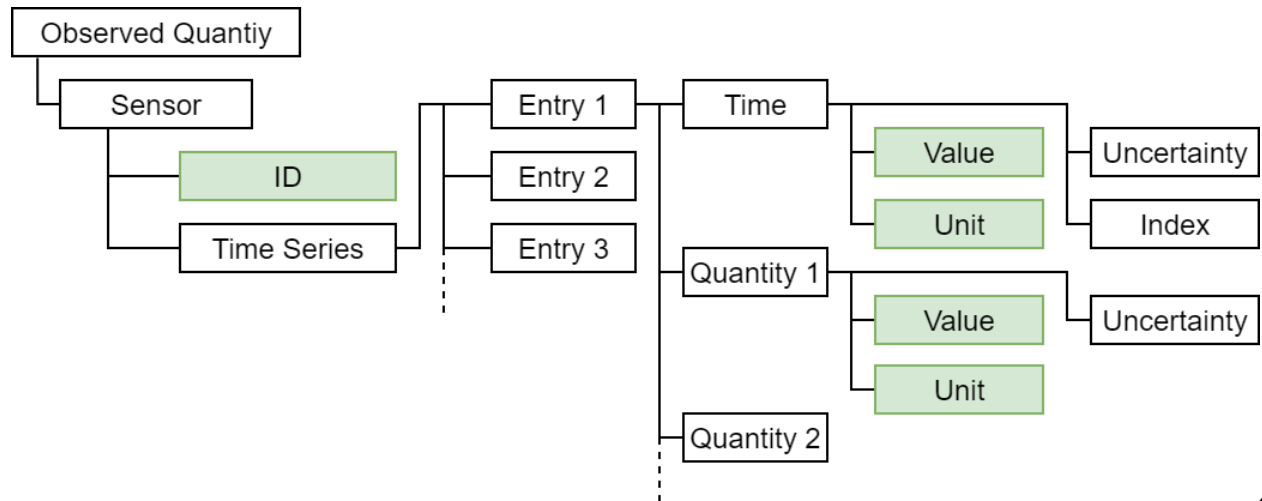
<sup>9</sup> <https://www.ptb.de>

<sup>10</sup> <https://www.ifm.eng.cam.ac.uk/>

<sup>11</sup> <https://met4fof.eu>

<sup>12</sup> <https://famous-project.eu>

<sup>13</sup> <https://devguide.python.org/#status-of-python-branches>



series metadata scheme illustration

The scheme contains all metadata to interpret the actual time and quantity values.

It consists of a dictionary containing the following keys with (default) values of the specified type:

```

metadata = {
  "device_id": string (default: ""),
  "time_name": string (default: "time"),
  "time_unit": string (default: "om:second"),
  "quantity_names": string or list of strings (default: ""),
  "quantity_units": string or list of strings (default: ""),
  "misc": optional, any other data you want to provide (default: None),
}

```

## 1.3 Example use

We illustrate the use of the scheme assuming you already have a project set up.

### 1.3.1 Installation

First you need to install the scheme with the usual command into your project's Python virtual environment:

```
pip install time-series-metadata
```

### 1.3.2 Import scheme

Inside your project's code import the scheme at the top of your module.

```
from time_series_metadata.scheme import MetaData
```

### 1.3.3 Assign initial values

After importing the package you can make use of it and assign initial values.



```

vs_description = MetaData(
    device_id="my_virtual_sensor",
    time_name="time",
    time_unit="s",
    quantity_names=("pressure_1", "pressure_2"),
    quantity_units=("Pa", "mPa"),
    misc="additional information"
)

```

### 1.3.4 Read out metadata

You can access the metadata as a whole or time and quantity metadata separately. Quantity metadata can be either accessed for all quantities at once or individually via index or name. If you do not specify name or index, the first's quantity metadata is returned. This might be especially convenient, if there is only one quantity.

```

>>> vs_description.metadata
{"device_id": "my_virtual_sensor", "time_name": "time", "time_unit": "s", "quantity_
↪names": ["pressure_1", "pressure_2"], "quantity_units": ["Pa", "mPa"], "misc": None}
>>> vs_description.time
{'time_name': 'time', 'time_unit': 's'}
>>> vs_description.quantities
{'quantity_names': ('pressure_1', 'pressure_2'), 'quantity_units': ('Pa', 'mPa')}
>>> vs_description.get_quantity(1)
{'quantity_names': 'pressure_2', 'quantity_units': 'mPa'}
>>> vs_description.get_quantity(name="pressure_1")
{'quantity_names': 'pressure_1', 'quantity_units': 'Pa'}
>>> vs_description.get_quantity()
{'quantity_names': 'pressure_1', 'quantity_units': 'Pa'}

```

## 1.4 Maintainers

The package is developed and maintained at the “Physikalisch-Technische Bundesanstalt” by Maximilian Gruber<sup>14</sup> and Björn Ludwig<sup>15</sup>.

<sup>14</sup> <https://github.com/mgrub>

<sup>15</sup> <https://github.com/BjoernLudwigPTB>



---

## MetaData - the metrologically enabled time-series metadata scheme

---

The module `time_series_metadata.scheme` contains the main class of the package. It provides the Python implementation of the scheme.

This module contains the following class:

- `MetaData`: Wrapper class for metrologically enabled time-series metadata

### 2.1 MetaData

```
class time_series_metadata.scheme.MetaData (device_id: str = "", time_name: str = 'time',
time_unit: str = 'om:second', quantity_names:
Union[str, Tuple[str, ...]] = "", quantity_units:
Union[str, Tuple[str, ...]] = "", misc: Op-
tional[Any] = None)
```

**get\_quantity** (*index*: int = 0, *name*: str = None) → Dict[KT, VT]

Return the metadata for one of the quantities

**Parameters**

- **index** (*int*, *optional*) – Index of the quantity in the initial tuple (default = 0). If *name* is set, *index* is ignored.
- **name** (*str*, *optional*) – Name of the quantity. If *name* is set, *index* is ignored.

**Returns metadata** – the metadata for the specified quantity

**Return type** dict

**metadata**

Return the metadata as a whole

**Returns metadata** – the metadata dictionary as a whole

**Return type** Dict

### **misc**

Return the additionally provided metadata

**Returns metadata** – all misc metadata key value pairs

**Return type** dict

### **quantities**

Return all quantities metadata

**Returns metadata** – all quantity metadata key value pairs

**Return type** dict

### **time**

Return the time metadata

**Returns metadata** – all time metadata key value pairs

**Return type** dict

## CHAPTER 3

---

### Indices and tables

---

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



**t**

`time_series_metadata.scheme`, 7





## G

`get_quantity()` (*time\_series\_metadata.scheme.MetaData* method), 7

## M

`MetaData` (class in *time\_series\_metadata.scheme*), 7

`metadata` (*time\_series\_metadata.scheme.MetaData* attribute), 7

`misc` (*time\_series\_metadata.scheme.MetaData* attribute), 7

## Q

`quantities` (*time\_series\_metadata.scheme.MetaData* attribute), 8

## T

`time` (*time\_series\_metadata.scheme.MetaData* attribute), 8

`time_series_metadata.scheme` (module), 7