
JSON Time Series (JTS) for Python

Slava Pisarevskiy

May 20, 2023

CONTENTS:

1	Library reference	1
2	JSON Time Series	3
2.1	Installation	3
2.2	Usage	3
2.3	TimeSeries	4
2.4	TsRecord	4
2.5	JTS Document	5
2.6	License	6
3	Indices and tables	7
	Index	9

LIBRARY REFERENCE

```
class json_timeseries.TsRecord(timestamp: datetime, value: float | str | int, quality: int = None, annotation: str = None)
```

A record of TimeSeries object

Parameters

- **timestamp** (datetime) – Timestamp
- **value** (Union[float, str, int]) – Value
- **quality** (int) – Quality
- **annotation** (str) – Annotation

```
class json_timeseries.TimeSeries(name: str, units: str = None, identifier: str = '6edea47f-824d-41cb-8936-31af5bf86ef0', data_type: str = 'NUMBER', records: List[TsRecord] | TsRecord = None)
```

TimeSeries object

Parameters

- **data_type** (str, optional) – Type of time series. E.g.: 'NUMBER', 'TEXT', 'TIME', 'COORDINATES'
- **records** (list, optional) – List of records
- **name** (str) – Time series name
- **identifier** (str, optional) – Time series ID. Autogenerated as UUID4 if not specified

```
insert(records: TsRecord | List[TsRecord])
```

Insert single or multiple records

```
toJSON() → str
```

Outputs formatted JSON

```
class json_timeseries.JtsDocument(series: List[TimeSeries] | TimeSeries = None, version: str = '1.0')
```

JTS document object

Raises

[TypeError] – [Value of 'series' must be types of TimeSeries or List[TimeSeries]]

```
addSeries(series: List[TimeSeries] | TimeSeries)
```

Add single or multiple TimeSeries

```
static fromJSON(json_str: str) → JtsDocument
```

Create a new jtsDocument from JSON

getSeries(*identifier: str*) → *TimeSeries*

Get series by id

toJSON() → dict

Output as dictionary of JSON structure

Returns

Python dictionary of JSON structure

Return type

dict

toJSONString() → str

Output as stringified JSON (json.dumps)

Returns

Output as stringified JSON

Return type

str

JSON TIME SERIES

JSON Time Series (JTS specification) handling Python library - Time Series data construction, manipulation and serialisation.

2.1 Installation

```
pip install json-timeseries
```

Import or require module

```
from json_timeseries import TsRecord, TimeSeries, JtsDocument
```

2.2 Usage

```
from json_timeseries import TsRecord, TimeSeries, JtsDocument
from datetime import datetime

# Create Time Series
timeseries1 = TimeSeries(identifier='series_1', name='Series 1', data_type='NUMBER',
    records=[
        TsRecord(**{"timestamp": datetime.now(), "value": '1.23', "quality": 192, "annotation": 'comment'}),
        TsRecord(**{"timestamp": datetime.now(), "value": '2.34', "quality": 245, "annotation": 'comment number 2'})])

timeseries2 = TimeSeries(identifier='series_2', name='Series 2', data_type='NUMBER',
    units="C",
    records=TsRecord(timestamp=datetime.now(), value=1.11,
    quality=111, annotation="comment ts2 111")
    )

# Add record(s)
timeseries1.insert(TsRecord(**{ timestamp: datetime.now(), value: 30 })))
```

(continues on next page)

(continued from previous page)

```
# Output in JSON Time Series document format
jts_doc = JtsDocument([timeseries1, timeseries2])
json_str = jts_doc.toJSONString()
```

2.3 TimeSeries

TimeSeries is a class for constructing and manipulating a single dataset.

```
from json_timeseries import TsRecord, TimeSeries
from datetime import datetime

time_series = TimeSeries(identifier='series_2', name='Series 2', data_type='NUMBER',
    ↳units="m/s",
    records=TsRecord(timestamp=datetime.now(), value=1.11,
    ↳quality=0, annotation="example comment")
    )
```

2.3.1 Options

Optionally provide configuration used for certain output formats such as JTS Document.

- **data_type**: data type of record **value** attribute. NUMBER | TEXT | TIME | COORDINATES
- **id**: string or number to uniquely identify the series to use instead of the automatically assigned id.
- **name**: string
- **units**: string
- **records**: list of data records

Alternatively set later:

```
time_series.data_type = 'NUMBER'
time_series.id = 'Series_1'
time_series.name = 'My Series'
time_series.units = 'm/s'
```

2.4 TsRecord

TsRecord is a class for constructing and manipulating a single record.

```
from json_timeseries import TsRecord
from datetime import datetime

ts_record1 = TsRecord(timestamp=datetime.now(), value=1.11, quality=0, annotation=
    ↳"example comment")
# Or as dict of parameters using ** operator
ts_record2 = TsRecord(**{"timestamp": datetime.now(), "value": 1.11, "quality": 0,
    ↳"annotation": 'example comment'})
```


2.4.1 Record attributes

Records require a timestamp and at least one attribute: value, quality or annotation

- **timestamp**: date object. Type of datetime. e.g. `datetime.now()`
- **value** (*optional*): number, string, date, null
- **quality** (*optional*): number (quality code) associated with value
- **annotation** (*optional*): string description or comment related to the record

2.4.2 Methods

See [full documentation](#).

2.4.3 Properties

See [full documentation](#).

2.5 JTS Document

`JtsDocument` is a class for outputting `TimeSeries` in [JSON Time Series](#) document format.

```
# Create a JTS Document from one or more timeseries
jts_document = JtsDocument(series=[timeseries1, timeseries2])
# Output series in JTS Document format
json_str = jts_document.toJSONString()
```

2.5.1 Options

- **series**: array of `TimeSeries` to include in JTS Document

2.5.2 Methods

See [full documentation](#).

2.5.3 Properties

See [full documentation](#).

2.6 License

MIT

INDICES AND TABLES

- `genindex`

INDEX

A

`addSeries()` (*json_timeseries.JtsDocument* method), 1

F

`fromJSON()` (*json_timeseries.JtsDocument* static method), 1

G

`getSeries()` (*json_timeseries.JtsDocument* method), 1

I

`insert()` (*json_timeseries.TimeSeries* method), 1

J

`JtsDocument` (class in *json_timeseries*), 1

T

`TimeSeries` (class in *json_timeseries*), 1

`toJSON()` (*json_timeseries.JtsDocument* method), 2

`toJSON()` (*json_timeseries.TimeSeries* method), 1

`toJSONString()` (*json_timeseries.JtsDocument* method), 2

`TsRecord` (class in *json_timeseries*), 1