
tabledata Documentation

Release 0.10.3

Tsuyoshi Hombashi

Feb 14, 2020

Table of Contents

| | | |
|----------|---|-----------|
| 1 | tabledata | 1 |
| 1.1 | Summary | 1 |
| 2 | Installation | 3 |
| 2.1 | Install from PyPI | 3 |
| 2.2 | Install from PPA (for Ubuntu) | 3 |
| 3 | Dependencies | 5 |
| 3.1 | Mandatory Python packages | 5 |
| 3.2 | Optional Python packages | 5 |
| 3.3 | Test dependencies | 5 |
| 4 | Reference | 7 |
| 4.1 | Data Structure | 7 |
| 4.1.1 | TableData | 7 |
| 4.2 | Exceptions | 10 |
| 5 | Indices and tables | 11 |
| 6 | Links | 13 |
| 7 | Indices and tables | 15 |
| | Index | 17 |

1.1 Summary

tabledata is a Python library to represent tabular data. Used for pytablewriter/pytablereader/SimpleSQLite.

2.1 Install from PyPI

```
pip install tabledata
```

2.2 Install from PPA (for Ubuntu)

```
sudo add-apt-repository ppa:thombashi/ppa  
sudo apt update  
sudo apt install python3-tabledata
```


Python 2.7+ or 3.5+

3.1 Mandatory Python packages

- `DataProperty` (Used to extract data types)
- `six`
- `typepy`

3.2 Optional Python packages

- **`loguru`**
 - Used for logging if the package installed
- **`pandas`**
 - required to get table data as a pandas data frame

3.3 Test dependencies

- `pytablewriter`
- `pytest`
- `tox`

4.1 Data Structure

4.1.1 TableData

class `tabledata.TableData` (*table_name*, *headers*, *rows*, *dp_extractor=None*, *type_hints=None*,
max_workers=None)

Class to represent a table data structure.

Parameters

- **table_name** (*str*) – Name of the table.
- **headers** (*list*) – Table header names.
- **rows** (*list*) – Data of the table.

`as_dataframe()`

Returns Table data as a `pandas.DataFrame` instance.

Return type `pandas.DataFrame`

Sample Code

```
from tabledata import TableData

TableData(
    "sample",
    ["a", "b"],
    [[1, 2], [3.3, 4.4]]
).as_dataframe()
```

Output

| | a | b |
|---|-----|-----|
| 0 | 1 | 2 |
| 1 | 3.3 | 4.4 |

Dependency Packages

- pandas

`as_dict()`

Returns Table data as a dict instance.

Return type dict

Sample Code

```
from tabledata import TableData

TableData(
    "sample",
    ["a", "b"],
    [[1, 2], [3.3, 4.4]]
).as_dict()
```

Output

```
{'sample': [OrderedDict([('a', 1), ('b', 2)]), OrderedDict([('a', 3.3), ('b', 4.4)])]}
```

`as_tuple()`

Returns Rows of the table.

Return type list of namedtuple

Sample Code

```
from tabledata import TableData

records = TableData(
    "sample",
    ["a", "b"],
    [[1, 2], [3.3, 4.4]]
).as_tuple()
for record in records:
    print(record)
```

Output

```
Row(a=1, b=2)
Row(a=Decimal('3.3'), b=Decimal('4.4'))
```

`column_dp_list`

`dp_extractor`

`equals` (*other*, *cmp_by_dp=False*)

`filter_column` (*patterns=None*, *is_invert_match=False*, *is_re_match=False*, *pattern_match=<PatternMatch.OR: 0>*)

static from_dataframe (*dataframe*, *table_name*=*u*)

Initialize TableData instance from a pandas.DataFrame instance.

Parameters

- **dataframe** (*pandas.DataFrame*) –
- **table_name** (*str*) – Table name to create.

has_value_dp_matrix

header_dp_list

headers

Get the table header names.

Returns Table header names.

Return type list or tuple

in_tabledata_list (*other*, *cmp_by_dp*=*False*)

is_empty ()

Returns True if the data *headers* or *value_matrix* is empty.

Return type bool

is_empty_header ()

Returns True if the data *headers* is empty.

Return type bool

is_empty_rows ()

Returns True if the tabular data has no rows.

Return type bool

num_columns

num_rows

Number of rows in the tabular data. None if the *rows* is neither list nor tuple.

Return type int

Type return

rows

Original rows of tabular data.

Returns Table rows.

Return type list or tuple

table_name

Name of the table. :rtype: str

Type return

transpose ()

validate_rows ()

Raises ValueError –

value_dp_matrix

DataProperty for table data. :rtype: list

Type return

value_matrix

Converted rows of tabular data.

Returns Table rows.

Return type list or tuple

4.2 Exceptions

exception tabledata.**NameValidationError**

Bases: exceptions.ValueError

Exception raised when a name is invalid.

exception tabledata.**InvalidTableNameError**

Bases: tabledata.error.NameValidationError

Exception raised when a table name is invalid.

exception tabledata.**InvalidHeaderNameError**

Bases: tabledata.error.NameValidationError

Exception raised when a table header name is invalid.

exception tabledata.**DataError**

Bases: exceptions.ValueError

Exception raised when data is invalid as tabular data.

CHAPTER 5

Indices and tables

- `genindex`

CHAPTER 6

Links

- [GitHub repository](#)
- [Issue tracker](#)
- [PyPI](#)
- [pip](#): A tool for installing Python packages

CHAPTER 7

Indices and tables

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

A

`as_dataframe()` (*tabledata.TableData* method), 7
`as_dict()` (*tabledata.TableData* method), 8
`as_tuple()` (*tabledata.TableData* method), 8

C

`column_dp_list` (*tabledata.TableData* attribute), 8

D

`DataError`, 10
`dp_extractor` (*tabledata.TableData* attribute), 8

E

`equals()` (*tabledata.TableData* method), 8

F

`filter_column()` (*tabledata.TableData* method), 8
`from_dataframe()` (*tabledata.TableData* static method), 8

H

`has_value_dp_matrix` (*tabledata.TableData* attribute), 9
`header_dp_list` (*tabledata.TableData* attribute), 9
`headers` (*tabledata.TableData* attribute), 9

I

`in_tabledata_list()` (*tabledata.TableData* method), 9
`InvalidHeaderNameError`, 10
`InvalidTableNameError`, 10
`is_empty()` (*tabledata.TableData* method), 9
`is_empty_header()` (*tabledata.TableData* method), 9
`is_empty_rows()` (*tabledata.TableData* method), 9

N

`NameValidationError`, 10
`num_columns` (*tabledata.TableData* attribute), 9

`num_rows` (*tabledata.TableData* attribute), 9

R

`rows` (*tabledata.TableData* attribute), 9

T

`table_name` (*tabledata.TableData* attribute), 9
`TableData` (*class in tabledata*), 7
`transpose()` (*tabledata.TableData* method), 9

V

`validate_rows()` (*tabledata.TableData* method), 9
`value_dp_matrix` (*tabledata.TableData* attribute), 9
`value_matrix` (*tabledata.TableData* attribute), 10