tabledata Documentation

Release 1.3.3

Tsuyoshi Hombashi

Sep 16, 2023

Table of Contents

CHAPTER 1

tabledata

1.1 Summary

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

CHAPTER 2

Installation

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
```

CHAPTER 3

Dependencies

- Python 3.7+
- Mandatory Python package dependencies (automatically installed)

3.1 Optional Python packages

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

CHAPTER 4

Reference

4.1 Data Structure

4.1.1 TableData

class tabledata.**TableData**(*table_name: Optional[str], headers: Sequence[str], rows: Sequence[T_co], dp_extractor: Optional[dataproperty._extractor.DataPropertyExtractor] = None, type_hints: Optional[Sequence[Union[str, Type[typepy.type._base.AbstractType], None]]] = None, max_workers: Optional[int] = None, max_precision: Optional[int] = None*)

Class to represent a table data structure.

Parameters

- **table_name** – Name of the table.
- **headers** – Table header names.
- **rows** – Data of the table.

as_dataframe() → pandas.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 (*default_key: str = 'table'*) → Dict[str, List[OrderedDict[str, Any]]]

Parameters **default_key** – Key of a returning dictionary when the `table_name` is empty.

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 () → Iterator[Tuple]

Returns Rows of the tuple.

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: tabledata._core.TableData*, *cmp_by_dp: bool = True*) → bool

filter_column(*patterns: Optional[str] = None*, *is_invert_match: bool = False*, *is_re_match: bool = False*, *pattern_match: tabledata._constant.PatternMatch = <PatternMatch.OR: 0>*) → tabledata._core.TableData

static from_dataframe(*dataframe: pandas.DataFrame*, *table_name: str = ''*, *type_hints: Optional[Sequence[Optional[Type[typepy.type._base.AbstractType]]]] = None*, *max_workers: Optional[int] = None*) → TableData

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

Table header names.

Type Sequence[str]

in_tabledata_list(*other: Sequence[TableData]*, *cmp_by_dp: bool = True*) → bool

is_empty() → bool

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

Return type bool

is_empty_header() → bool

bool: `True` if the data *headers* is empty.

is_empty_rows() → bool

Returns `True` if the tabular data has no rows.

Return type bool

max_workers

num_columns

num_rows

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

Type Optional[int]

rows

Original rows of tabular data.

Type Sequence

table_name

Name of the table.

Type str

transpose() → tabledata._core.TableData

validate_rows() → None

Raises **ValueError** –

value_dp_matrix

DataProperty for table data.

Type DataPropertyMatrix

value_matrix
Converted rows of tabular data.

Type DataPropertyMatrix

4.2 Exceptions

exception tabledata.**NameValidationError**
Bases: 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: ValueError

Exception raised when data is invalid as tabular data.

CHAPTER 5

Indices and tables

- genindex

CHAPTER 6

Links

- GitHub repository
- Issue tracker
- pip: A tool for installing Python packages

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

Index