xltable Documentation

Release 0.1

Renshaw Bay

Jan 29, 2018

Contents

A package for writing excel worksheets with formulas and styles.

CHAPTER 1

Introduction

xltable is an API for writing tabular data and charts to Excel. It is not a replacement for other Excel writing packages such as `xlsxwriter`, `xlwt` or `pywin32`. Instead it uses those packages as a back end to write the Excel files (or to write to Excel directly in the case of `pywin32`) and provides a higer level abstraction that allows the programmer to deal with tables of data rather than worry about writing individual cells.

The main feature that makes *xltable* more useful than just writing the Excel files directly is that it can handle tables with formulas that relate to cells in the workbook *without* having to know in advance where those tables will be placed on a worksheet. Only when all the tables have been added to the workbook and the workbook is being written are formulas resolved to their final cell addresses.

Tables of data are constructed using `pandas.DataFrame` objects. These can contain formulas relating to columns or cells in the same table or other tables in the same workbook.

As well as writing tables to Excel, *xltable* can also write charts using tables as source data.

Integrating xltable into Excel can be done using PyXLL, https://www.pyxll.com. PyXLL embeds a Python interpreter within Excel and makes it possible to use Excel as a front end user interface to Python code. For example, you could configure a custom ribbon control for users to run Python reports and have the results written back to Excel.

CHAPTER 2

Example

Write a dataframe with a formula to Excel:

```
from xltable import *
import pandas as pa

# create a dataframe with three columns where the last is the sum of the first two
dataframe = pa.DataFrame({
        "col_1": [1, 2, 3],
        "col_2": [4, 5, 6],
        "col_3": Cell("col_1") + Cell("col_2"),
}, columns=["col_1", "col_2", "col_3"])

# create the named xltable Table instance
table = Table("table", dataframe)

# create the Workbook and Worksheet objects and add table to the sheet
sheet = Worksheet("Sheet1")
sheet.add_table(table)

workbook = Workbook("example.xlsx")
workbook.add_sheet(sheet)

# write the workbook to the file (requires xlsxwriter)
workbook.to_xlsx()
```

CHAPTER 3

Classes

class xltable.**Workbook**(*filename=None*, *worksheets=[]*)
A workbook is an ordered collection of worksheets.

Once all worksheets have been added the workbook can be written out or the worksheets can be iterated over, and any expressions present in the tables of the worksheets will be resolved to absolute worksheet/cell references.

Parameters

- **filename** (*str*) – Filename the workbook will be written to.
- **worksheets** (*list*) – List of *xltable.Worksheet* instances.

add_sheet(*worksheet*)
Adds a worksheet to the workbook.

to_xlsx(***kwargs*)
Write workbook to a .xlsx file using xlsxwriter. Return a xlsxwriter.workbook.Workbook.

Parameters **kwargs** – Extra arguments passed to the xlsxwriter.Workbook

constructor.

to_excel(*xl_app=None*, *resize_columns=True*)

class xltable.**Worksheet**(*name='Sheet1'*)
A worksheet is a collection of tables placed at specific locations.

Once all tables have been placed the worksheet can be written out or the rows can be iterated over, and any expressions present in the tables will be resolved to absolute cell references.

Parameters **name** (*str*) – Worksheet name.

name
Worksheet name

add_table(*table*, *row=None*, *col=0*, *row_spaces=1*)
Adds a table to the worksheet at (row, col). Return the (row, col) where the table has been put.

Parameters

- **table** (xltable.Table) – Table to add to the worksheet.
- **row** (*int*) – Row to start the table at (defaults to the next free row).
- **col** (*int*) – Column to start the table at.
- **row_spaces** (*int*) – Number of rows to leave between this table and the next.

add_chart(*chart*, *row*, *col*)

Adds a chart to the worksheet at (row, col).

Parameters

- **Chart** (xltable.Chart) – chart to add to the workbook.
- **row** (*int*) – Row to add the chart at.

get_table_pos(*tablename*)

Parameters **tablename** (*str*) – Name of table to get position of.

Returns Upper left (row, col) coordinate of the named table.

get_table(*tablename*)

Parameters **tablename** (*str*) – Name of table to find.

Returns A *xltable.Table* instance from the table name.

next_row

Row the next table will start at unless another row is specified.

class xltable.**Table**(*name*, *dataframe*, *include_columns=True*, *include_index=False*, *style='default'*, *column_styles={}*, *column_widths={}*, *row_styles={}*, *header_style=None*, *index_style=None*)

Represents of table of data to be written to Excel, and may include :py:class:‘xltable.Expression‘s that will be converted into Excel formulas when the table’s position is fixed.

Parameters

- **name** (*str*) – Name of the table so it can be referenced by other tables and charts.
- **dataframe** (*pandas.DataFrame*) – Dataframe containing the data for the table.
- **include_columns** (*bool*) – Include the column names when outputting.
- **include_index** (*bool*) – Include the index when outputting.
- **style** (xltable.TableStyle) – Table style, or one of the named styles ‘default’ or ‘plain’.
- **column_styles** (xltable.CellStyle) – Dictionary of column names to styles or named styles.
- **column_widths** (*dict*) – Dictionary of column names to widths.
- **header_style** (xltable.CellStyle) – Style or named style to use for the cells in the header row.
- **index_style** (xltable.CellStyle) – Style or named style to use for the cells in the index column.

Named table styles:

- default: blue stripes
- plain: no style

Named cell styles:

- pct: pecentage with two decimal places.
- iso-date: date in YYYY-MM-DD format.
- 2dp: two decimal places.
- 2dpc: thousand separated number to two decimal places.

get_data(*workbook*, *row*, *col*, *formula_values={}*)

Returns 2d numpy array for this table with any formulas resolved to the final excel formula.

Parameters

- **workbook** (xltable.Workbook) – Workbook the table has been added to.
- **row** (*int*) – Row where the table will start in the sheet (used for resolving formulas).
- **col** (*int*) – Column where the table will start in the sheet (used for resolving formulas).
- **formula_values** – dict to add pre-calculated formula values to (keyed by row, col).

class xltable.**Chart**(*type*, *subtype=None*, *title=None*, *legend_position=None*, *x_axis=None*, *y_axis=None*, *show_blanks=None*, *width=480*, *height=288*)

Chart objects reference data from Table instances and are written to Excel worksheets as Excel charts.

Parameters

- **type** (*str*) – Chart type (see below).
- **subtype** (*str*) – Chart sub type (see below).
- **title** (*str*) – Chart title
- **legend_position** (*str*) – right (default), left, top, bottom or 'none' for no legend.
- **width** (*int*) – Chart width.
- **height** (*int*) – Chart height.

Chart types and sub-types:

- **area:**
  - stacked
  - percent_stacked
- **bar:**
  - stacked
  - perecent_stacked
- **column:**
  - stacked
  - perecent_stacked
- line
- **scatter:**
  - straight_with_markers
  - straight

  - smooth_with_markers
  - smooth
- stock
- **radar:**
  - with_markers
  - filled

add_series(*values*, ***kwargs*)

Adds a series to the chart.

Parameters

- **values** – A *xltable.Expression* object that evaluates to the data series.
- **categories** – A *xltable.Expression* object that evaluates to the data series.
- **name** – Name to show in the legend for the series
- **line** – Line style, eg {'color': 'blue', 'width': 3.25} or {'none': True}
- **marker** – dict specifying how the markers should look, eg {type: square}.
- **trendline** – dict specifying how the trendline should be drawn, eg {type: linear}.

class xltable.**ArrayFormula**(*name*, *formula*, *width*, *height*, *value=None*, *include_columns=False*, *include_index=False*, *style='default'*, *column_styles={}*, *column_widths={}*)

Represents an array formula to be written to Excel.

Subclass of *xltable.Table*.

Parameters

- **name** (*str*) – Name of table so it can be referenced by other tables and charts.
- **formula** (xltable.Formula) – Array formula.
- **width** (*int*) – Number of columns.
- **height** (*int*) – Number of row.
- **value** (*pandas.DataFrame*) – Precalculated formula result to save in the workbook.
- **include_columns** (*bool*) – Include the column names when outputting *value*.
- **include_index** (*bool*) – Include the index when outputting *value*.
- **style** (xltable.TableStyle) – Table style, or one of the named styles 'default' or 'plain'.
- **column_styles** (xltable.CellStyle) – Dictionary of column names to styles or named styles.
- **column_widths** (*dict*) – Dictionary of column names to widths.

class xltable.**Expression**(*value=None*)

Base class for all worksheet expressions.

Expressions are used to build formulas referencing ranges in the worksheet by labels which are resolved to cell references when the worksheet is written out.

Expressions may be combined using binary operators.

class xltable.**ArrayExpression**(*expr*)
Wraps an expression in an array formula (ie. surrounds it with {})

Parameters **expr** (xltable.Expression) – Expression to be wrapped

class xltable.**Cell**(*col*, *row=None*, *row_offset=0*, *table=None*, *col_fixed=None*, *row_fixed=None*, ***kwargs*)
Reference to a cell in a table.

Parameters

- **col** – Column label this refers to.
- **row** – Row label this refers to, or None to use the current row.
- **row_offset** – Offset from the row, used when resolving.
- **table** – Name of table the column is in, if not in the same table this expression is in. Use "%s!%s" % (worksheet.name, table.name) if refering to a table in another worksheet
- **col_fixed** – If True when converted to an address the column will be fixed.
- **row_fixed** – If True when converted to an address the row will be fixed.

class xltable.**Column**(*col*, *include_header=False*, *table=None*, *col_fixed=True*, *row_fixed=True*, ***kwargs*)
Reference to a column in a table.

Parameters

- **col** – Column label this refers to.
- **include_header** – True if this expression should include the column header.
- **table** – Name of table the column is in, if not in the same table this expression is in. Use "%s!%s" % (worksheet.name, table.name) if refering to a table in another worksheet
- **col_fixed** – If True when converted to an address the column will be fixed.
- **row_fixed** – If True when converted to an address the row will be fixed.

class xltable.**Index**(*include_header=False*, *table=None*, *col_fixed=True*, *row_fixed=True*, ***kwargs*)
Reference to a table's index.

Parameters

- **include_header** – True if this expression should include the index header.
- **table** – Name of table that owns the index, if not the table this expression is in. Use "%s!%s" % (worksheet.name, table.name) if refering to a table in another worksheet
- **col_fixed** – If True when converted to an address the column will be fixed.
- **row_fixed** – If True when converted to an address the row will be fixed.

class xltable.**Range**(*left_col*, *right_col*, *top_row=None*, *bottom_row=None*, *include_header=True*, *table=None*, *col_fixed=True*, *row_fixed=True*, ***kwargs*)
Reference to a range in a table.

Parameters

- **left_col** – Left most column label this refers to.
- **right_col** – Right most column label this refers to.
- **top_row** – Top most row label, or None to select from the top of the table.
- **bottom_row** – Bottom most row label, or None to select to the bottom of the table.

- **include_header** – Include table header in the range.
- **table** – Name of table the column is in, if not in the same table this expression is in. Use "%s!%s" % (worksheet.name, table.name) if refering to a table in another worksheet
- **col_fixed** – If True when converted to an address the column will be fixed.
- **row_fixed** – If True when converted to an address the row will be fixed.

class xltable.**Formula**(*name*, **args*, ***kwargs*)

Formula expression.

E.g. to create a formula like "=SUMPRODUCT(a, b)" where a and b are columns in a table you would do:

```
formula = Formula("SUMPRODUCT", Column("col_a"), Column("col_b"))
```

Parameters

- **name** – Name of Excel function, eg "SUMPRODUCT".
- **args** – Expressions to use as arguments to the function.

class xltable.**TableStyle**(*stripe_colors=(15397370*, *16777215)*, *border=None*)

Style to be applied to a table.

Parameters **stripe_colors** (*tuple*) – Background cell colors to use as RGB values, e.g. 0xFF0000 for red.

class xltable.**CellStyle**(*is_percentage=None*, *decimal_places=None*, *date_format=None*, *thousands_sep=None*, *excel_number_format=None*, *bold=None*, *size=None*, *text_color=None*, *bg_color=None*, *text_wrap=None*, *border=None*, *align=None*, *valign=None*)

Style to be applied to a cell or range of cells.

Parameters

- **is_percentage** (*bool*) – True if the cell value is a percentage.
- **decimal_places** (*int*) – Number of decimal places to display the cell value to.
- **date_format** (*str*) – Format to use for date values (use Python date format, e.g. '%Y-%m-%d').
- **thousands_sep** (*bool*) – True to display numbers with thousand separator.
- **excel_number_format** (*str*) – Excel number format; overrides other numeric settings (eg thousands_sep).
- **bold** (*bool*) – True to make cells bold.
- **size** (*int*) – Text size, or use one of the string size aliases x-small, small, normal, large, x-large or xx-large.
- **text_color** (*int*) – Text color as an RGB value, e.g. 0xFF0000 for red.
- **bg_color** (*int*) – Background color as an RGB value, e.g. 0xFF0000 for red.
- **border** – Can only be used when writing using xlsxwriter. Can be an xlsxwriter border style index (integer) or a dictionary of styles keyed by border position (top, bottom, left, right).

class xltable.**Value**(*value*, *style=None*)

Value wrapper that can be used in a table to add a style.

Parameters

- **value** – Value that will be written to the cell.
- **xltable.CellStyle** – Style to be applied to the cell.

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

Python Module Index

x

Index