
TableReport Documentation

Release 0.1

Jeffrey

Jun 04, 2020

Table of Contents

1	Example	3
1.1	Basic	3
1.2	Style	3
1.3	Column Selector	4
1.4	Row Selector	5
1.5	Cell Selector	5
1.6	Merge	6
1.7	Group	7
1.8	Summary	7
1.9	Horizontal Summary	8
1.10	Complex	9
2	API Reference	11
2.1	Table	11
2.2	shortcut	12
3	Indices and tables	13
	Index	15

TableReport is a python library for making table report. Now supports exporting to Excel.

CHAPTER 1

Example

1.1 Basic

	A	B	C	D
1	HEADER1	HEADER2	HEADER3	HEADER4
2	One	A	1	2
3	One	A	2	3
4	One	B	3	4
5				

```
from tablereport import Table
from tablereport.shortcut import write_to_excel

table = Table(header=[['HEADER1', 'HEADER2', 'HEADER3', 'HEADER4']],
               body=[['One', 'A', 1, 2],
                     ['One', 'A', 2, 3],
                     ['One', 'B', 3, 4]])

write_to_excel('basic.xlsx', table)
```

1.2 Style

```
from tablereport import Table, Style
from tablereport.shortcut import write_to_excel

table_style = Style({
    'background_color': 'fff0f0f0',
```

(continues on next page)

	A	B	C	D	
1	TEST				
2	HEADER1	HEADER2	HEADER3	HEADER4	
3	One	A	1	2	
4	One	A	2	3	
5	One	B	3	4	
6					

(continued from previous page)

```

})
title_style = Style({
    'background_color': 'ff87cefa',
    'font_weight': 'bold'
}, extend=table_style)

header_style = Style({
    'background_color': 'ff87cefa',
}, extend=table_style)
table = Table(header=[('TEST', title_style), None, None, None],
              [('HEADER1', header_style),
               ('HEADER2', header_style),
               ('HEADER3', header_style),
               ('HEADER4', header_style)]],
              body=[['One', 'A', 1, 2],
                    ['One', 'A', 2, 3],
                    ['One', 'B', 3, 4]],
              style=table_style)

write_to_excel('style.xlsx', table)

```

1.3 Column Selector

	A	B	C	D	
1	HEADER1	HEADER2	HEADER3	HEADER4	
2	One	A	1	2	
3	One	A	2	3	
4	One	B	3	4	
5					

```

from tablereport import Table, ColumnSelector, Style
from tablereport.shortcut import write_to_excel

table = Table(header=[('HEADER1', 'HEADER2', 'HEADER3', 'HEADER4')],
              body=[['One', 'A', 1, 2],
                    ['One', 'A', 2, 3],

```

(continues on next page)

(continued from previous page)

```

        ['One', 'B', 3, 4]])

style = Style({
    'background_color': 'fff0f0f0',
})
areas = table.body.select(ColumnSelector(lambda col: col % 2))
areas.set_style(style)
write_to_excel('column_selector.xlsx', table)

```

1.4 Row Selector

	A	B	C	D
1	HEADER1	HEADER2	HEADER3	HEADER4
2	One	A	1	2
3	One	A	2	3
4	One	B	3	4
5	One	B	1	2

```

from tablereport import Table, RowSelector, Style
from tablereport.shortcut import write_to_excel

header_style = Style({
    'background_color': 'FF87CEFA',
})
even_row_style = Style({
    'background_color': 'FFF0F0F0',
})

table = Table(header=[['HEADER1', 'HEADER2', 'HEADER3', 'HEADER4']],
              body=[['One', 'A', 1, 2],
                    ['One', 'A', 2, 3],
                    ['One', 'B', 3, 4],
                    ['One', 'B', 1, 2], 1])

table.header.set_style(header_style)
rows = table.body.select(RowSelector(lambda line: not line % 2))
rows.set_style(even_row_style)
write_to_excel('row_selector.xlsx', table)

```

1.5 Cell Selector

```

from tablereport import Table, Style, ColumnSelector, CellSelector
from tablereport.shortcut import write_to_excel

good_score_style = Style({

```

(continues on next page)

	A	B	C	D	
1	HEADER1	HEADER2	HEADER3	HEADER4	
2	One	A	90	70	
3	One	A	50	40	
4	One	B	93	59	
5	Two	A	78	23	
6	Two	B	28	66	
7					

(continued from previous page)

```

    'background_color': 'ff00cc33',
})
bad_score_style = Style({
    'background_color': 'ffcc0000',
})

table = Table(header=[['HEADER1', 'HEADER2', 'HEADER3', 'HEADER4']],
              body=[['One', 'A', 90, 70],
                    ['One', 'A', 50, 40],
                    ['One', 'B', 93, 59],
                    ['Two', 'A', 78, 23],
                    ['Two', 'B', 28, 66]])

area = table.body.select(ColumnSelector(lambda col: col == 3, width=2)).one()
good_score_cells = area.select(Selector(lambda cell: cell.value >= 90))
bad_score_cells = area.select(Selector(lambda cell: cell.value < 60))
good_score_cells.set_style(good_score_style)
bad_score_cells.set_style(bad_score_style)

```

1.6 Merge

	A	B	C	D	
1	HEADER1	HEADER2	HEADER3	HEADER4	
2	One	A	1	2	
3		A	2	3	
4		B	3	4	
5					

```

from tablereport import Table, ColumnSelector
from tablereport.shortcut import write_to_excel

table = Table(header=[['HEADER1', 'HEADER2', 'HEADER3', 'HEADER4']],
              body=[['One', 'A', 1, 2],
                    ['One', 'A', 2, 3],
                    ['One', 'B', 3, 4]])

```

(continues on next page)

(continued from previous page)

```
column = table.body.select(ColumnSelector(lambda col: col == 1)).one()
column.merge()
write_to_excel('merge.xlsx', table)
```

1.7 Group

	A	B	C	D	
1	HEADER1	HEADER2	HEADER3	HEADER4	
2	One	A	1	2	
3		A	2	3	
4		B	3	4	
5	Two	A	1	2	
6		B	2	3	
7					

```
from tablereport import Table, ColumnSelector
from tablereport.shortcut import write_to_excel

table = Table(header=[['HEADER1', 'HEADER2', 'HEADER3', 'HEADER4']],
              body=[['One', 'A', 1, 2],
                    ['One', 'A', 2, 3],
                    ['One', 'B', 3, 4],
                    ['Two', 'A', 1, 2],
                    ['Two', 'B', 2, 3]])

column = table.body.select(ColumnSelector(lambda col: col == 1)).one()
column.group().merge()
write_to_excel('group.xlsx', table)
```

1.8 Summary

	A	B	C	D	
1	HEADER1	HEADER2	HEADER3	HEADER4	
2	One	A	1	2	
3	One	A	2	3	
4	One	B	3	4	
5	Total		6	9	
6					

```

from tablereport import Table, Style
from tablereport.shortcut import write_to_excel

table = Table(header=[['HEADER1', 'HEADER2', 'HEADER3', 'HEADER4']],
              body=[['One', 'A', 1, 2],
                    ['One', 'A', 2, 3],
                    ['One', 'B', 3, 4]])

style = Style({
    'background_color': 'ffe6e6e6',
})
table.body.summary(label='Total', label_span=2, label_style=style,
                  value_style=style)
write_to_excel('summary.xlsx', table)

```

1.9 Horizontal Summary

	A	B	C	D	E
1	HEADER1	HEADER2	HEADER3	HEADER4	TOTAL
2	One	A	1	2	3
3	One	A	2	3	5
4	One	B	3	4	7
5	Two	A	1	2	3
6	Two	B	2	3	5
7					

```

from tablereport import Table, Style, ColumnSelector
from tablereport.shortcut import write_to_excel

table = Table(header=[['HEADER1', 'HEADER2', 'HEADER3', 'HEADER4']],
              body=[['One', 'A', 1, 2],
                    ['One', 'A', 2, 3],
                    ['One', 'B', 3, 4],
                    ['Two', 'A', 1, 2],
                    ['Two', 'B', 2, 3]])

header_style = Style({
    'background_color': 'ff87cefa',
})
summary_style = Style({
    'background_color': 'ffe6e6e6',
})

areas = table.select(ColumnSelector(lambda col: col == 3, width=2))
summary_style = Style({
    'background_color': 'ffe6e6e6',
})
areas.summary(label_span=1, label='TOTAL', location='right',
              value_style=summary_style, label_style=header_style)
table.header.set_style(header_style)
write_to_excel('horizontal_summary.xlsx', table)

```

1.10 Complex

	A	B	C	D
1	TEST			
2	HEADER1	HEADER2	HEADER3	HEADER4
3	One	A	1	2
4		A	2	3
5		B	3	4
6		Total	6	9
7	Two	A	1	2
8		B	2	3
9		Total	3	5
10	Total		9	14
11				

```

from tablereport import Table, ColumnSelector, Style
from tablereport.shortcut import write_to_excel

title_style = Style({
    'font_size': 15,
    'background_color': 'ff87cefa',
    'font_weight': 'blod'
})

header_style = Style({
    'background_color': 'ff87cefa',
})

left_total_style = Style({
    'background_color': 'fff0f0f0',
})

bottom_total_style = Style({
    'background_color': 'ffe6e6e6',
})

table = Table(header=[['TEST', None, None, None],
                      ['HEADER1', 'HEADER2', 'HEADER3', 'HEADER4']],
              body=[['One', 'A', 1, 2],
                    ['One', 'A', 2, 3],
                    ['One', 'B', 3, 4],
                    ['Two', 'A', 1, 2],
                    ['Two', 'B', 2, 3]])

table.header[0].set_style(title_style)
table.header[1].set_style(header_style)

```

(continues on next page)

(continued from previous page)

```
column = table.body.select(ColumnSelector(lambda col: col == 1)).one()
column.group().merge().left.summary(label_span=1, label='Total',
                                     label_style=left_total_style,
                                     value_style=left_total_style)

table.summary(label_span=2, label='Total',
              label_style=bottom_total_style,
              value_style=bottom_total_style)

write_to_excel('complex.xlsx', table)
```

2.1 Table

class `tablereport.tablereport.Table` (*header=None, body=None, style=None*)

Table is the core class of TableReport.

A table looks like a nested list. Each inner list represents a row of the table, and each row contains some cells:

```
[
    [Cell('header1'), Cell('header2')],
    [Cell(1), Cell(2)],
    [Cell(3), Cell(4)]
]
```

We can create this by the following ways, and each element in a row will be auto wrapped into a cell:

```
table = Table(
    header=[['header1', 'header2']],
    body=[[1, 2], [3, 4]]
)
```

All Cells in a table has a style attribute, the default value of which can be set by `Style` argument, and we can also separately set the style attribute of a cell as below:

```
table = Table(
    header=[('header1', style), 'header2'],
    body=[[1, 2], [3, 4]]
)
```

An import thing is that `None` in a row will be specially handled. `None` will be used to auto merge cells. A sample could explain this:

```
table = Table(header=[['test', None], ['header1', 'header2']],
              body=[[1, 2], ])
```

This will create a table as below. This feature is usually used for custom table header:

```
[[Cell('test', width=2), None],  
 [Cell('header1'), Cell('header2')],  
 [Cell(1), Cell(2)]]
```

2.2 shortcut

2.2.1 write_to_excel

`tablereport.shortcut.write_to_excel(filename, table, position=(0, 0))`
write table into excel.

If the file does not exist, a new file will be created. If the file has already existed, the file will be rewritten.

By default, the table will be written in default worksheet at position (0,0). You can change the position by setting `position` argument.

CHAPTER 3

Indices and tables

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

T

Table (*class in `tablereport.tablereport`*), [11](#)

W

write_to_excel() (*in module `tablereport.shortcut`*),
[12](#)