

---

# **Isogeo - XLSX Exporter Documentation**

***Release 1.3.2***

**Isogeo**

**May 14, 2020**



**CONTENTS:**

<b>1</b>	<b>Indices and tables</b>	<b>3</b>
1.1	Package modules . . . . .	3
	<b>Python Module Index</b>	<b>11</b>
	<b>Index</b>	<b>13</b>



**Author** Isogeo

**Source code** <https://github.com/Isogeo/isogeotxlsx/>

**Issues** <https://github.com/Isogeo/isogeotxlsx/issues>

Updated: 2020-05-14



## INDICES AND TABLES

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

### 1.1 Package modules

#### 1.1.1 `isogeotoxlsx`

##### `isogeotoxlsx` package

This package is used to export Isogeo metadata into Excel workbooks using the Isogeo Python SDK and OpenPyXL.

##### Subpackages

##### `isogeotoxlsx.i18n` package

##### Submodules

##### `isogeotoxlsx.i18n.english` module

Matching table between Isogeo metadata model, the extended attributes for Excel output and columns titles in FRENCH.

##### `isogeotoxlsx.i18n.french` module

Matching table between Isogeo metadata model, the extended attributes for Excel output and columns titles in FRENCH.

### isogeotoxlsx.matrix package

`isogeotoxlsx.matrix.ColumnPattern`  
alias of `isogeotoxlsx.matrix.Column`

#### Submodules

##### isogeotoxlsx.matrix.attributes module

Matching table of Excel columns for Feature Attributes analysis.

##### isogeotoxlsx.matrix.raster module

Matching table between Isogeo metadata model and Excel columns for Isogeo to Office.

##### isogeotoxlsx.matrix.resource module

Matching table between Isogeo metadata model and Excel columns for Isogeo to Office.

##### isogeotoxlsx.matrix.service module

Matching table between Isogeo metadata model and Excel columns for Isogeo to Office.

##### isogeotoxlsx.matrix.vector module

Matching table between Isogeo metadata model and Excel columns for Isogeo to Office.

### isogeotoxlsx.utils package

#### Submodules

##### isogeotoxlsx.utils.formatter module

**class** `isogeotoxlsx.utils.formatter.Formatter` (*lang='FR', output\_type='Excel'*)  
Bases: `object`

Metadata formatter to avoid repeat operations on metadata during export in different formats.

##### Parameters

- **lang** (*str*) – selected language
- **output\_type** (*str*) – name of output type to format for. Defaults to 'Excel'
- **default\_values** (*tuple*) – values used to replace missing values. Structure:  
( str\_for\_missing\_strings\_and\_integers, str\_for\_missing\_dates  
)



**conditions** (*md\_cgus*)

Render input metadata CGUs as a new list.

**Parameters** **md\_cgus** (*list*) – list of conditions extracted from an Isogeo metadata

**Return type** *list*

**Example**

```
# make a search including conditions
search = isogeo.search(include=("conditions",))

# parse results
for md in search.results:
    # load metadata as object
    md = Metadata.clean_attributes(md)

    # format conditions
    cgus_out = formatter.conditions(md.conditions)
```

**frequency\_as\_explicit\_str** (*update\_frequency\_code*)

Transform 'updateFrequency' code value as an explicit string. See: <https://github.com/isogeo/export-xlsx-py/issues/8>

**Parameters** **update\_frequency\_code** (*str*) – update frequency as stored in Isogeo API

**Returns** update frequency as explicit string.

**Return type** *str*

**Example**

```
>>> print(frequency_as_explicit_str("P1D"))
>>> "Every 1 day(s) "
```

**limitations** (*md\_limitations*)

Render input metadata limitations as a new list.

**Parameters** **md\_limitations** (*dict*) – input dictionary extracted from an Isogeo metadata

**Return type** *list*

**specifications** (*md\_specifications*)

Render input metadata specifications as a new list.

**Parameters** **md\_specifications** (*dict*) – input dictionary extracted from an Isogeo metadata

**Return type** *list*

**isogeotxlsx.utils.stats module**

Statistics Calculator. Perform statistics analysis.

Author: Isogeo

Python: 3.7.x Created: 14/08/2014

**class** `isogeotxlsx.utils.stats.Stats` (*lang='fr'*)

Bases: `object`

Perform statistics calculations and make Excel charts.

**attributes** (*ws\_attributes*, *all\_attributes*)

Perform feature attributes analysis and write results into the wanted worksheet.

**Parameters**

- **ws\_attributes** (*Worksheet*) – sheet of a Workbook to write analysis
- **all\_attributes** (*list*) – list of all feature attributes. It's a list of dicts.

**li\_data\_formats** = []

**li\_dates\_md\_created** = []

**li\_dates\_md\_modified** = []

**line\_dates** (*ws*, *li\_dates\_md\_created*=None, *li\_dates\_md\_modified*=None, *cell\_start\_table*='O1',  
*cell\_start\_chart*='S1')

Calculates metadata creation and modification dates repartition and add a Line chart to the wanted sheet of Workbook.

**Parameters**

- **ws** (*Worksheet*) – sheet of a Workbook to write analysis
- **li\_dates\_md\_created** (*list*) – list of metadatas' creation dates. If not specified, the class attribute will be used instead.
- **li\_dates\_md\_modified** (*list*) – list of metadatas' modification dates. If not specified, the class attribute will be used instead.
- **cell\_start\_table** (*str*) – cell of the sheet where to start writing table
- **cell\_start\_chart** (*str*) – cell of the sheet where to start writing the chart

**md\_empty\_fields** = {}

**md\_tags\_occurrences** = {}

**md\_types\_repartition** = {}

**pie\_formats** (*ws*, *li\_formats*=None, *cell\_start\_table*='A20', *cell\_start\_chart*='D20')

Calculates metadata formats repartition and add a Pie chart to the wanted sheet of Workbook.

**Parameters**

- **ws** (*Worksheet*) – sheet of a Workbook to write analysis
- **li\_formats** (*list*) – list of all formats labels. If not specified, the class attribute will be used instead
- **cell\_start\_table** (*str*) – cell of the sheet where to start writing table
- **cell\_start\_chart** (*str*) – cell of the sheet where to start writing the chart

**pie\_types** (*ws*, *types\_counters*=None, *cell\_start\_table*='A1', *cell\_start\_chart*='D1')

Calculates metadata types repartition and add a Pie chart to the wanted sheet of Workbook.

**Parameters**

- **ws** (*Worksheet*) – sheet of a Workbook to write analysis
- **types\_counters** (*dict*) – dictionary of types/count. If not specified, the class attribute will be used instead
- **cell\_start\_table** (*str*) – cell of the sheet where to start writing table
- **cell\_start\_chart** (*str*) – cell of the sheet where to start writing the chart

## Submodules

### isogeotoxlsx.isogeo2xlsx module

Get metadatas from Isogeo and store it into a Excel worksheet.

```
class isogeotoxlsx.isogeo2xlsx.Isogeo2xlsx (lang='FR', url_base_edit="",
                                           url_base_view="", **kwargs)
    Bases: openpyxl.workbook.workbook.Workbook
```

Used to store Isogeo API results into an Excel worksheet (.xlsx)

#### Parameters

- **lang** (*str*) – selected language for output
- **url\_base\_edit** (*str*) – base url to format edit links (basically app.isogeo.com)
- **url\_base\_view** (*str*) – base url to format view links (basically open.isogeo.com)

```
column_width (ws, columns)
```

Set the width of the columns of the passed worksheet.

#### Parameters

- **ws** (*Worksheet*) – worksheet into write headers
- **columns** (*ColumnPattern*) – column table

```
headers_writer (ws, columns)
```

Writes the headers from a columns ref table to a worksheet.

#### Parameters

- **ws** (*Worksheet*) – worksheet into write headers
- **columns** (*ColumnPattern*) – column table

```
launch_analysis ()
```

Launches special analysis, using the stats submodule.

```
row_height (ws, from_row=2, height=35)
```

Set the height of the rows of the passed worksheet.

#### Parameters

- **ws** (*Worksheet*) – worksheet into write headers
- **from\_row** (*int*) – row to start from. Default to '2' = ignoring headers.
- **height** (*int*) – fixed height to apply. Default to 35.

```
set_worksheets (auto=None, vector=1, raster=1, service=1, resource=1, dashboard=0, attributes=0, fillfull=0, inspire=0)
```

Adds new sheets depending on present metadata types in isogeo API search tags.

#### Parameters

- **auto** (*list*) – typically auto=search\_results.get('tags').keys()
- **vector** (*bool*) – add vector sheet
- **raster** (*bool*) – add raster sheet
- **service** (*bool*) – add service sheet
- **resource** (*bool*) – add resource sheet

- **dashboard** (*bool*) – add dashboard sheet
- **attributes** (*bool*) – add attributes sheet - only if vector is True too
- **fillfull** (*bool*) – add fillfull sheet
- **inspire** (*bool*) – add inspire sheet

**store\_md\_generic** (*md, ws, idx*)

Exports generic metadata attributes into Excel worksheet with some dynamic adaptations based on meta-data type.

**Parameters**

- **md** (*Metadata*) – metadata object to export
- **ws** (*Worksheet*) – Excel worksheet to store the exported info
- **idx** (*int*) – row index in the worksheet

**store\_metadatas** (*metadata, share=None*)

Write metadata into the worksheet.

**Parameters**

- **metadata** (*Metadata*) – metadata object to write
- **share** (*Share*) – share to use to build the OpenCatalog URL

**styling\_cells** (*ws, columns*)

Applies the referenced style to the cells of a column.

**Parameters**

- **ws** (*Worksheet*) – worksheet into write headers
- **columns** (*ColumnPattern*) – column table

**tunning\_worksheets** (*excluded\_sheets='dashboard'*)

Applies some adjustments to the sheets of the workbook: filters, frozen panels, print settings, etc.

**Parameters** **excluded\_sheets** (*tuple*) – sheets name to be excluded from the tunning.

## isogeotxlsx.isogeoFromxlsx module

Make Metadatas from Excel worksheet like those returned by Isogeo2xlsx.

**class** isogeotxlsx.isogeoFromxlsx.**IsogeoFromxlsx** (*file\_path="", lang='fr'*)

Bases: object

Used to read Isogeo Metadata stored into an Excel worksheet (.xlsx)

**Parameters**

- **file\_path** (*Path*) – the path of xlsx file to read
- **lang** (*str*) – selected language for input

**build\_event** (*event\_date, kind*)

Build an Event instance from a date as string

**build\_index\_dict** (*work\_sheet, ref\_dict*)

Build a dictionary where keys are Metadata (isogeo-pysdk objet) attributes and values are corresponding column index

**build\_inspireTh** (*inspireTh\_value*)

Build a dict of INSPIRE themes from a string where each keyword label is delimited with ‘;’, where keys are theme label and keys are theme uuid into Isogeo database

**build\_keywords** (*keywords\_value*)

Build a list of Keyword instances from a string where each keyword label is delimited with ‘;’

**build\_list** (*text*)

Build a list from a string where different values are separated by ‘;’

**build\_md\_conditions** (*conditions\_value*)

Build a list of Conditions instances from a string where each condition is delimited with ‘;’ retrieving licenses in the list of Licenses previously loaded if the excel file contains a ‘Licenses’ work sheet

**build\_md\_contacts** (*contacts\_value*)

Build a list of Contact instances from a string where each contact email is delimited with ‘;’ retrieving theme in the list of Contacts previously loaded if the excel file contains a ‘Contacts’ work sheet

**build\_md\_limitations** (*limitations\_value*)

Build a list of Limitations instances from a string where each limitation is delimited with ‘;’ retrieving directives in the list of Directive previously loaded if the excel file contains a ‘Directives’ work sheet

**build\_md\_spec** (*specs\_value*)

Build a list of Specifications instances from a string where each specification is delimited with ‘;’ retrieving specifications uuid in the list of Specifiactions previously loaded if the excel file contains a ‘Specifications’ work sheet

**read\_file** ()

Simple method to read the xlsx file’s content

**retrieve\_sub\_ressources** (*sub\_ressource*)**retrieve\_vector\_metadataas** ()

Method to retrieve Isogeo vectors metadataas from appropriate sheet and store theme as isogeo-pysdk models objects into class attributes



## PYTHON MODULE INDEX

### i

- `isogeotoxlsx`, 1
- `isogeotoxlsx.i18n`, 3
- `isogeotoxlsx.i18n.english`, 3
- `isogeotoxlsx.i18n.french`, 3
- `isogeotoxlsx.isogeo2xlsx`, 7
- `isogeotoxlsx.isogeoFromxlsx`, 8
- `isogeotoxlsx.matrix`, 4
- `isogeotoxlsx.matrix.attributes`, 4
- `isogeotoxlsx.matrix.raster`, 4
- `isogeotoxlsx.matrix.resource`, 4
- `isogeotoxlsx.matrix.service`, 4
- `isogeotoxlsx.matrix.vector`, 4
- `isogeotoxlsx.utils`, 4
- `isogeotoxlsx.utils.formatter`, 4
- `isogeotoxlsx.utils.stats`, 5





## A

`attributes()` (*isogeotools.utils.stats.Stats* method),  
5

## B

`build_event()` (*isogeotools.isogeotools.Isogeotools*  
method), 8

`build_index_dict()` (*isogeotools.isogeotools.Isogeotools*  
method), 8

`build_inspireTh()` (*isogeotools.isogeotools.Isogeotools*  
method), 8

`build_keywords()` (*isogeotools.isogeotools.Isogeotools*  
method), 9

`build_list()` (*isogeotools.isogeotools.Isogeotools*  
method), 9

`build_md_conditions()` (*isogeotools.isogeotools.Isogeotools*  
method), 9

`build_md_contacts()` (*isogeotools.isogeotools.Isogeotools*  
method), 9

`build_md_limitations()` (*isogeotools.isogeotools.Isogeotools*  
method), 9

`build_md_spec()` (*isogeotools.isogeotools.Isogeotools*  
method), 9

## C

`column_width()` (*isogeotools.isogeotools.Isogeotools*  
method), 7

`ColumnPattern` (in module *isogeotools.matrix*), 4

`conditions()` (*isogeotools.utils.formatter.Formatter*  
method), 4

## F

`Formatter` (class in *isogeotools.utils.formatter*), 4

`frequency_as_explicit_str()` (*isogeotools.utils.formatter.Formatter*  
method), 5

## H

`headers_writer()` (*isogeotools.isogeotools.Isogeotools*  
method), 7

## I

`Isogeotools` (class in *isogeotools.isogeotools*), 7  
`Isogeotools` (class in *isogeotools.isogeotools*), 8

`isogeotools`  
module, 1, 3

`isogeotools.i18n`  
module, 3

`isogeotools.i18n.english`  
module, 3

`isogeotools.i18n.french`  
module, 3

`isogeotools.isogeotools`  
module, 7

`isogeotools.isogeotools`  
module, 8

`isogeotools.matrix`  
module, 4

`isogeotools.matrix.attributes`  
module, 4

`isogeotools.matrix.raster`  
module, 4

`isogeotools.matrix.resource`  
module, 4

`isogeotools.matrix.service`  
module, 4

`isogeotools.matrix.vector`  
module, 4

`isogeotools.utils`  
module, 4

`isogeotools.utils.formatter`  
module, 4

`isogeotools.utils.stats`

module, 5

## L

launch\_analisis() (isogeotoxlsx.isogeo2xlsx.Isogeo2xlsx method), 7

li\_data\_formats (isogeotoxlsx.utils.stats.Stats attribute), 6

li\_dates\_md\_created (isogeotoxlsx.utils.stats.Stats attribute), 6

li\_dates\_md\_modified (isogeotoxlsx.utils.stats.Stats attribute), 6

limitations() (isogeotoxlsx.utils.formatter.Formatter method), 5

line\_dates() (isogeotoxlsx.utils.stats.Stats method), 6

## M

md\_empty\_fields (isogeotoxlsx.utils.stats.Stats attribute), 6

md\_tags\_occurrences (isogeotoxlsx.utils.stats.Stats attribute), 6

md\_types\_repartition (isogeotoxlsx.utils.stats.Stats attribute), 6

module

- isogeotoxlsx, 1, 3
- isogeotoxlsx.il18n, 3
- isogeotoxlsx.il18n.english, 3
- isogeotoxlsx.il18n.french, 3
- isogeotoxlsx.isogeo2xlsx, 7
- isogeotoxlsx.isogeoFromxlsx, 8
- isogeotoxlsx.matrix, 4
- isogeotoxlsx.matrix.attributes, 4
- isogeotoxlsx.matrix.raster, 4
- isogeotoxlsx.matrix.resource, 4
- isogeotoxlsx.matrix.service, 4
- isogeotoxlsx.matrix.vector, 4
- isogeotoxlsx.utils, 4
- isogeotoxlsx.utils.formatter, 4
- isogeotoxlsx.utils.stats, 5

## P

pie\_formats() (isogeotoxlsx.utils.stats.Stats method), 6

pie\_types() (isogeotoxlsx.utils.stats.Stats method), 6

## R

read\_file() (isogeotoxlsx.isogeoFromxlsx.IsogeoFromxlsx method), 9

retrieve\_sub\_ressources() (isogeotoxlsx.isogeoFromxlsx.IsogeoFromxlsx method), 9

retrieve\_vector\_metadatas() (isogeotoxlsx.isogeoFromxlsx.IsogeoFromxlsx method), 9

row\_height() (isogeotoxlsx.isogeo2xlsx.Isogeo2xlsx method), 7

## S

set\_worksheets() (isogeotoxlsx.isogeo2xlsx.Isogeo2xlsx method), 7

specifications() (isogeotoxlsx.utils.formatter.Formatter method), 5

Stats (class in isogeotoxlsx.utils.stats), 5

store\_md\_generic() (isogeotoxlsx.isogeo2xlsx.Isogeo2xlsx method), 8

store\_metadatas() (isogeotoxlsx.isogeo2xlsx.Isogeo2xlsx method), 8

styling\_cells() (isogeotoxlsx.isogeo2xlsx.Isogeo2xlsx method), 8

## T

tunning\_worksheets() (isogeotoxlsx.isogeo2xlsx.Isogeo2xlsx method), 8