

---

# **Isogeo - XLSX Exporter Documentation**

*Release 1.3.2*

**Isogeo**

**May 14, 2020**



**CONTENTS:**

**1 Indices and tables** **3**  
    1.1 Package modules . . . . . **3**  
**Python Module Index** **11**  
**Index** **13**



**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 isogeotxlsx

##### **isogeotxlsx package**

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

##### **Subpackages**

##### **isogeotxlsx.i18n package**

##### **Submodules**

##### **isogeotxlsx.i18n.english module**

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

##### **isogeotxlsx.i18n.french module**

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

### isogeotxlsx.matrix package

isogeotxlsx.matrix.**ColumnPattern**  
alias of isogeotxlsx.matrix.Column

#### Submodules

##### isogeotxlsx.matrix.attributes module

Matching table of Excel columns for Feature Attributes analysis.

##### isogeotxlsx.matrix.raster module

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

##### isogeotxlsx.matrix.resource module

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

##### isogeotxlsx.matrix.service module

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

##### isogeotxlsx.matrix.vector module

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

### isogeotxlsx.utils package

#### Submodules

##### isogeotxlsx.utils.formatter module

**class** isogeotxlsx.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

**isogeotoxlsx.utils.stats module**

Statistics Calculator. Perform statistics analysis.

Author: Isogeo

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

**class** isogeotoxlsx.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, at-  
tributes=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 metadata 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\_metadata** (*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\_metadatas** ()

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



## PYTHON MODULE INDEX

### i

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





- A**
- attributes() (*isogeotxlsx.utils.stats.Stats* method), 5
- B**
- build\_event() (*isogeotxlsx.isogeoFromxlsx.IsogeoFromxlsx* method), 8
  - build\_index\_dict() (*isogeotxlsx.isogeoFromxlsx.IsogeoFromxlsx* method), 8
  - build\_inspireTh() (*isogeotxlsx.isogeoFromxlsx.IsogeoFromxlsx* method), 8
  - build\_keywords() (*isogeotxlsx.isogeoFromxlsx.IsogeoFromxlsx* method), 9
  - build\_list() (*isogeotxlsx.isogeoFromxlsx.IsogeoFromxlsx* method), 9
  - build\_md\_conditions() (*isogeotxlsx.isogeoFromxlsx.IsogeoFromxlsx* method), 9
  - build\_md\_contacts() (*isogeotxlsx.isogeoFromxlsx.IsogeoFromxlsx* method), 9
  - build\_md\_limitations() (*isogeotxlsx.isogeoFromxlsx.IsogeoFromxlsx* method), 9
  - build\_md\_spec() (*isogeotxlsx.isogeoFromxlsx.IsogeoFromxlsx* method), 9
- C**
- column\_width() (*isogeotxlsx.isogeo2xlsx.Isogeo2xlsx* method), 7
  - ColumnPattern (*in module isogeotxlsx.matrix*), 4
  - conditions() (*isogeotxlsx.utils.formatter.Formatter* method), 4
- F**
- Formatter (*class in isogeotxlsx.utils.formatter*), 4
- H**
- frequency\_as\_explicit\_str() (*isogeotxlsx.utils.formatter.Formatter* method), 5
  - headers\_writer() (*isogeotxlsx.isogeo2xlsx.Isogeo2xlsx* method), 7
- I**
- Isogeo2xlsx (*class in isogeotxlsx.isogeo2xlsx*), 7
  - IsogeoFromxlsx (*class in isogeotxlsx.isogeoFromxlsx*), 8
  - isogeotxlsx
    - module, 1, 3
  - isogeotxlsx.i18n
    - module, 3
    - isogeotxlsx.i18n.english
      - module, 3
    - isogeotxlsx.i18n.french
      - module, 3
    - isogeotxlsx.isogeo2xlsx
      - module, 7
    - isogeotxlsx.isogeoFromxlsx
      - module, 8
    - isogeotxlsx.matrix
      - module, 4
      - isogeotxlsx.matrix.attributes
        - module, 4
      - isogeotxlsx.matrix.raster
        - module, 4
      - isogeotxlsx.matrix.resource
        - module, 4
      - isogeotxlsx.matrix.service
        - module, 4
      - isogeotxlsx.matrix.vector
        - module, 4
    - isogeotxlsx.utils
      - module, 4
      - isogeotxlsx.utils.formatter
        - module, 4
      - isogeotxlsx.utils.stats
        - module, 4

module, 5

## L

launch\_analisis() (*isogeotxlsx.isogeo2xlsx.Isogeo2xlsx method*), 7

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

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

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

limitations() (*isogeotxlsx.utils.formatter.Formatter method*), 5

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

## M

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

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

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

module

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

## P

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

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

## R

read\_file() (*isogeotxlsx.isogeoFromxlsx.IsogeoFromxlsx method*), 9

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

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

row\_height() (*isogeotxlsx.isogeo2xlsx.Isogeo2xlsx method*), 7

## S

set\_worksheets() (*isogeotxlsx.isogeo2xlsx.Isogeo2xlsx method*), 7

specifications() (*isogeotxlsx.utils.formatter.Formatter method*), 5

Stats (*class in isogeotxlsx.utils.stats*), 5

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

store\_metadatas() (*isogeotxlsx.isogeo2xlsx.Isogeo2xlsx method*), 8

styling\_cells() (*isogeotxlsx.isogeo2xlsx.Isogeo2xlsx method*), 8

## T

tunning\_worksheets() (*isogeotxlsx.isogeo2xlsx.Isogeo2xlsx method*), 8