Isogeo - XLSX Exporter Documentation

Release 1.2.4

Isogeo

Jan 03, 2020

Contents:

1 Indices and tables **3**
1.1 Cookbook 3
1.2 Package modules 4

Python Module Index **11**

Index **13**

Author Isogeo

Source code https://github.com/Isogeo/isogeotoxlsx/

Issues https://github.com/Isogeo/isogeotoxlsx/issues

Updated: 2020-01-03

CHAPTER 1

Indices and tables

- genindex
- modindex
- search

1.1 Cookbook

1.1.1 Usage

IsogeoToXlsx processes a Isogeo Search as defined in the Isogeo Python SDK.

1. Store your secret as environment variables (using an `.env` file for example)
2. Authenticate

```
# import
from isogeo_pysdk import Isogeo
# API client
isogeo = Isogeo(
    auth_mode="group",
    client_id=ISOGEO_API_GROUP_CLIENT_ID,
    client_secret=ISOGEO_API_GROUP_CLIENT_SECRET,
    auto_refresh_url="{}/oauth/token".format(ISOGEO_ID_URL),
    platform=ISOGEO_PLATFORM,
)

# getting a token
isogeo.connect()
```

3. Make a search:

```
search = isogeo.search(include="all", page_size=100)
# close session
isogeo.close()
```

4. Export:

```
# import
from io import BytesIO
from isogeotoxlsx import Isogeo2xlsx
# instanciate the final workbook
out_workbook = Isogeo2xlsx(
    lang=isogeo.lang,
    url_base_edit=isogeo.app_url,
    url_base_view=isogeo.oc_url,
    write_only=True
)
# add needed worksheets
out_workbook.set_worksheets(
    auto=search.tags.keys(),  # create the relevant sheets according to the␣
↪metadata types
    dashboard=True  # add a dashboard sheet
)

# map search results and store method
for md in map(Metadata.clean_attributes, search.results):
    out_workbook.store_metadatas(md)

# save as file
out_workbook.save("./isogeo_export_to_xlsx.xlsx")

# or save in a memory object
mem_virtual_workbook = BytesIO()
out_workbook.save(mem_virtual_workbook)
```

1.2 Package modules

1.2.1 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 analisis.

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`

isogeotoxlsx.utils.stats module

Statistics Calculator. Perform statistics analisis.

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 analisis and write results into the wanted worksheet.

Parameters

- **ws_attributes** (*Worksheet*) – sheet of a Workbook to write analisis
- **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 analisis
- **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_occurences = {}

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 analisis
- **li_formats** (*list*) – list of all formats labels. If not specified, the class attribute will be used instaed
- **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 analisis
- **types_counters** (*dict*) – dictionary of types/count. If not specified, the class attribute will be used instaed

- **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_analisis()

Launches special analisis, 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 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_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.

Python Module Index

i

isogeotoxlsx, 1
isogeotoxlsx.i18n, 4
isogeotoxlsx.i18n.english, 4
isogeotoxlsx.i18n.french, 5
isogeotoxlsx.isogeo2xlsx, 8
isogeotoxlsx.matrix, 5
isogeotoxlsx.matrix.attributes, 5
isogeotoxlsx.matrix.raster, 5
isogeotoxlsx.matrix.resource, 5
isogeotoxlsx.matrix.service, 5
isogeotoxlsx.matrix.vector, 5
isogeotoxlsx.utils, 5
isogeotoxlsx.utils.formatter, 5
isogeotoxlsx.utils.stats, 6

Index

A

`attributes()` (*isogeotoxlsx.utils.stats.Stats method*), 7

C

`column_width()` (*isogeotoxlsx.isogeo2xlsx.Isogeo2xlsx method*), 8
`ColumnPattern` (*in module isogeotoxlsx.matrix*), 5
`conditions()` (*isogeotoxlsx.utils.formatter.Formatter method*), 6

F

`Formatter` (*class in isogeotoxlsx.utils.formatter*), 5
`frequency_as_explicit_str()` (*isogeotoxlsx.utils.formatter.Formatter method*), 6

H

`headers_writer()` (*isogeotoxlsx.isogeo2xlsx.Isogeo2xlsx method*), 8

I

`Isogeo2xlsx` (*class in isogeotoxlsx.isogeo2xlsx*), 8
`isogeotoxlsx` (*module*), 1, 4
`isogeotoxlsx.i18n` (*module*), 4
`isogeotoxlsx.i18n.english` (*module*), 4
`isogeotoxlsx.i18n.french` (*module*), 5
`isogeotoxlsx.isogeo2xlsx` (*module*), 8
`isogeotoxlsx.matrix` (*module*), 5
`isogeotoxlsx.matrix.attributes` (*module*), 5
`isogeotoxlsx.matrix.raster` (*module*), 5
`isogeotoxlsx.matrix.resource` (*module*), 5
`isogeotoxlsx.matrix.service` (*module*), 5
`isogeotoxlsx.matrix.vector` (*module*), 5
`isogeotoxlsx.utils` (*module*), 5
`isogeotoxlsx.utils.formatter` (*module*), 5
`isogeotoxlsx.utils.stats` (*module*), 6

L

`launch_analisis()` (*isogeotoxlsx.isogeo2xlsx.Isogeo2xlsx method*), 8
`li_data_formats` (*isogeotoxlsx.utils.stats.Stats attribute*), 7
`li_dates_md_created` (*isogeotoxlsx.utils.stats.Stats attribute*), 7
`li_dates_md_modified` (*isogeotoxlsx.utils.stats.Stats attribute*), 7
`limitations()` (*isogeotoxlsx.utils.formatter.Formatter method*), 6
`line_dates()` (*isogeotoxlsx.utils.stats.Stats method*), 7

M

`md_empty_fields` (*isogeotoxlsx.utils.stats.Stats attribute*), 7
`md_tags_occurences` (*isogeotoxlsx.utils.stats.Stats attribute*), 7
`md_types_repartition` (*isogeotoxlsx.utils.stats.Stats attribute*), 7

P

`pie_formats()` (*isogeotoxlsx.utils.stats.Stats method*), 7
`pie_types()` (*isogeotoxlsx.utils.stats.Stats method*), 7

R

`row_height()` (*isogeotoxlsx.isogeo2xlsx.Isogeo2xlsx method*), 8

S

`set_worksheets()` (*isogeotoxlsx.isogeo2xlsx.Isogeo2xlsx method*), 8
`specifications()` (*isogeotoxlsx.utils.formatter.Formatter method*), 6

`Stats` (*class in isogeotoxlsx.utils.stats*), 6
`store_md_generic()` (*isogeotoxlsx.isogeo2xlsx.Isogeo2xlsx method*), 9
`store_metadatas()` (*isogeotoxlsx.isogeo2xlsx.Isogeo2xlsx method*), 9
`styling_cells()` (*isogeotoxlsx.isogeo2xlsx.Isogeo2xlsx method*), 9

T

`tunning_worksheets()` (*isogeotoxlsx.isogeo2xlsx.Isogeo2xlsx method*), 9