xl2roefact python library API Reference

app_cli

app_cli: the command line application for all xl2roefact functionalities.

Identification:

- copyright: (c) 2023 RENWare Software Systems
- author: Petre Iordanescu (petre.iordanescu@gmail.com)

about

```
@app_cli.command()
def about()
```

provide a short application description.

settings

```
@app_cli.command()
def settings(rules: Annotated[
    bool,
    typer.
    Option("--rules", "-r", help="show settings recommended update rules"),
] = False)
```

display application configuration parameters and settings that are subject to be changed by user.

Arguments:

- `rules` - show recommended rules to follow when change application configurable settings (available in both RO & EN languages). Defaults to `False`.

xl2json

```
@app_cli.command()
def xl2json(
    file_name: Annotated[
        str, typer.Argument(
            help="files to process (wildcards allowed)")] = "*.xlsx",
    files_directory: Annotated[
        Path,
        typer.Option(
            "--files-directory",
            "-d",
            exists=False,
            file_okay=False,
            dir_okay=True,
            writable=True,
            readable=True,
            resolve_path=True,
            help=
            "directory to be used to look for Excel files (if default directory does not exists
will consider current directory instead)."
        ),
    ] = "invoice_files/",
    verbose: Annotated[
        bool,
        typer.
        Option("--verbose", "-v", help="show detailed processing messages"),
    ] = False)
```

extract data from an Excel file (save data to JSON format file with the same name as original file but `.json` extension).

Arguments:

- `file_name` - files to process (wildcards allowed).
- `files_directory` - directory to be used to look for Excel files. Defaults to `invoice_files/`. NOTE: if default directory does not exists will consider current directory instead
- `verbose` - show detailed processing messages". Defaults to `False`.

called_when_no_command

```
@app_cli.callback(invoke_without_command=True)
def called_when_no_command(
    ctx: typer.Context,
    version: Annotated[
        bool,
        typer.Option("--version", help="show application version"),
    ] = False)
```

function called when no command is invoked and to provide only application version (for external users to test it!).

run

NOTE: for `run` "reason to be" as copy of `app_cli` see iss `0.1.22b 240216piu_a`

chkisld

chkisld: modul de verificare a starii de incarcare a unei facturi emise

Identification:

- code-name: `chkisld`
- copyright: (c) 2023 RENWare Software Systems
- author: Petre Iordanescu (petre.iordanescu@gmail.com)

Specifications:

- document cerinte initiale: `110-SRE-api_to_roefact_requirements.md` section `Componenta xl2roefact`
- INTRARI: fisier `f-XLSX` sau numarul / cheia / codul facturii
- IESIRI: valoarea echivalent `TRUE` daca factura a fost deja incarcata sau valoare echivalent `FALSE` daca factura nu a fost incarcata

chkxml

chkxml: modul de validare a facturii in sistemul ANAF E-Factura

Identification:

- code-name: `chkxml`
- copyright: (c) 2023 RENWare Software Systems
- author: Petre Iordanescu (petre.iordanescu@gmail.com)

Specifications:

- document cerinte initiale: `110-SRE-api_to_roefact_requirements.md` section `Componenta xl2roefact`
- INTRARI: fisier `f-XML`
- IESIRI: raport cu eventualele erori de validare

config_settings

Configuration and setting parameters.

(en-us) README before making changes:

- each parameter has a short help (lines starting with `#` character) - read it before changing that parameter
- do not change parametrs name as specified before equal (`=`) sign
- lists are enclosed in sqaured brackets (`[...]`) and items are separated by comma character (`,`)
- strings are enclosed in `"` characters
- if you want to clear a list (for example you do not wants any options inside) just let it as `<PARAMETR NAME> = []` - do not drop that parameter
- do not add supplementary parameters, they will not be used without software changes (also risk to induce potential errors)
- for calendaristic dates Excel cells use `date` format and change it as display option to show wanted format

(ro) README inainte de a face modificari:

- fiecare parametru are un hep scurt (liniile ce incep cu caracterul `#`) - citi-l inainte de a modifica uun parametru
- nu schimbati numele parametrilor asa cum este el specificat inainte de semnul egal (`=`)
- listele sunt incluse intre paranteze drepte (`[...]`) si elementele lor sunt separate prin caracterul virgula (`,`)
- sirurile de caractere sunt incluse intre ghilimele (caracterul `"`)
- daca doriti stergerea unei listei (de ex daca nu doriti nici o optiune pentru acea lista) doar lasati acel parametru cu valoarea `[]` - nu stergeti in nici un caz acel parametru
- nu adaugati parametrii suplimentari (altii decit cei specificati aici), acestia nu vor fi utilizati fara a modifica aplicatia (de asemenea riscati sa induceti erori in cod)
- pentru datele calendaristice in celulul Excel a se utiliza formatul standard de data (`date`) si modificati formatul de afisare in formatul dorit pe factura tiparibila

DEFAULT_SUPPLIER_COUNTRY

NOTE: "pattern-uri" (sabloane) de identificare si regasire a datelor folositi de

__ comanda `xl2json` reprezentind functionalitatea de extragere a datelor din Excel si exportul lor in formatul JSON (modulul `rdinv)__

---

ldxml

ldxml: modul de incarcare a facturii in sistemul ANAF E-Factura

Identification:

- code-name: `ldxml`

- copyright: (c) 2023 RENWare Software Systems
- author: Petre Iordanescu (petre.iordanescu@gmail.com)

Specifications:

- document cerinte initiale: `110-SRE-api_to_roefact_requirements.md` section `Componenta xl2roefact`
- INTRARI: fisier `f-XML`
- IESIRI: raport cu validarea si identificatorul incarcarii

libutils

general utilities library for all `xl2roefact` components and modules.

Identification:

- code-name: `libutils`
- copyright: (c) 2023, 2024 RENWare Software Systems
- author: Petre Iordanescu (petre.iordanescu@gmail.com)

Components:

- `dict_sum_by_key(dict, str) -> float`: Sum a dictionary for a given key at all depth levels
- `find_str_in_list(list, list) -> int`: Search more strings (ie, a list) in list of strings
- `invoice_taxes_summary(list[dict]) -> dict`: Calculates invoice taxes summary as required by ROefact requirements
- `isnumber(str) -> bool`: Test a string if it could be used as number (int or float)

invoice_taxes_summary

```
def invoice_taxes_summary(invoice_lines: list[dict]) -> list
```

Calculates invoice taxes summary as required by ROefact requirements.

Arguments:

- `invoice_lines` - section with item lines from 'big' invoice dictionary

Returns:

`list` usable for "cac_TaxSubtotal" key

dict_sum_by_key

```
def dict_sum_by_key(search_dict: dict | list[dict], sum_key: str) -> float
```

Sum all dictionary (or list off dictionaries) items, at all levels, for a given key.

Arguments:

- `search_dict` - dictionary to be searched for
- `sum_key` - key to be searched

Returns:

`float` with required sum

isnumber

```
def isnumber(a_string: str) -> bool
```

test if a string is valid as any kind of number.

Arguments:

- `a_string` - input string.

Returns:

- `True` - if input string is valid as any kind of number, orherwise `False`.

find_str_in_list

```
def find_str_in_list(list_of_str_to_find: list, list_to_search: list) -> int
```

find a substring from `list_of_str_to_find` in elements of `list_to_search`.

Arguments:

- `list_of_str_to_find` - list of strings to search for.
- `list_to_search` - liste where to search for substrings.

Returns:

- `index` - the index of list item which contains `str_to_find` (first found) or `None` if not found.

rdinv

rdinv: modul de procesare a fisierului Excel ce contine factura si colectare a datelor aferente.

Formatul acceptat fisier Excel este `XLSX`.

Identification:

- code-name: `rdinv`

- copyright: (c) 2023 RENWare Software Systems
- author: Petre Iordanescu (petre.iordanescu@gmail.com)

Specifications:

- document cerinte initiale: `110-SRE-api_to_roefact_requirements.md` section `Componenta xl2roefact`
- INTRARI: fisier format XLSX ce contine factura emisa (cod: `f-XLSX`)
- IESIRI: fisier format JSON imagine a datelor facturii (cod: `f-JSON`)

rdinv

```
def rdinv(file_to_process: str,
          invoice_worksheet_name: str = None,
          *,
          debug_info: bool = False) -> dict
```

read Excel file for invoice data.

Produce a dictionary structure + JSON file with all data regarding read invoice: canonical KV data, meta data, map to convert to XML and original Excel data.

Arguments:

- `file_to_process` - the invoice file (exact file with path).
- `invoice_worksheet_name` - the worksheet containing invoice, optional, defaults to first found worksheet.
- `debug_info` - key only, show debugging information, default `False`.

Returns:

- `dict` - the invoice extracted information from Excel file as `dict(Invoice: dict, meta_info: dict, excel_original_data: dict)` `TODO` subject of documentation update.

NOTE ref important variables: * `db: pylightxl object`: EXCEL object with invoice (as a whole) * `ws: pylightxl object`: WORKSHEET object with invoice

get_excel_data_at_label

```
def get_excel_data_at_label(pattern_to_search_for: list[str],
                            worksheet: xl.Database.ws,
                            area_to_scan: list[list[int]] = None,
                            targeted_type: Callable = str,
                            down_search_try: bool = True) -> dict
```

get "one key Excel values", like invoice number or invoice issue date.

Arguments:

- `pattern_to_search_for` - for example for inv number, will pass the `PATTERN_FOR_INVOICE_NUMBER_LABEL`.
- `worksheet` - the worksheet containing invoice (as object of `pyxllight` library).

- `area_to_scan` - area of cells to be searched, default whole worksheet.
- `targeted_type` - what type expect (will try to convert to, if cannot will return str), default `str`.
- `down_search_try` - establish if DOWN search method is tried, default `True`.

Returns:

`None` if not found OR `dictionary` containing: * `"value": int | float | str` - the value found covenrted to requested `targeted_type` if possible or `str` otherwise; if "out of space" then returns `None` * `"location": (row, col)` - adrees of cell where found value

Notes:

- normal scan order is 1.RIGHT, 2.DOWN (if allowed), 3.IN-LABEL only in given area and pattern.

mk_kv_invoice_items_area

```
def mk_kv_invoice_items_area(invoice_items_area_xl_format) -> dict
```

transform `invoice_items_area` in "canonical JSON format" (as kv pairs).

Arguments:

- `invoice_items_area_xl_format` - invoice items area in Excel format (ie, DataFrame with row, col, data).

Returns:

- `invoice_items_area_xl_format` - dictionary with invoice items in Excel format (ie, rows, columns).

Notes:

- for ROefact XML model (& plan) see `invoice_files/__model_test_factura_generat_anaf.xml`.

get_invoice_items_area

```
def get_invoice_items_area(worksheet, invoice_items_area_marker,
                           wks_name) -> dict
```

get invoice for `invoice_items_area`, process it and return its Excel format.

Process steps & notes: * find invoice items subtable. * clean invoice items subtable. * extract relevenat data. * NOTE: all Excel cell addresses are in `(row, col)` format (ie, Not Excel format like "A:26, C:42, ...")

Arguments:

- `worksheet` - the worksheet containing invoice (as object of `pyxllight` library).
- `invoice_items_area_marker` - string with exact marker of invoice items table.
- `NOTE` - this is the UPPER-LEFT corner and is determined before calling this procedure.
- `wks_name` - the wroksheet name (string) of the `worksheet` object.

Returns:

- `invoice_items_area` - dictionary with invoice items in Excel format (ie, rows, columns).

wrxml

wrxml: modul de generare a fisierului format XML

Identification:

- code-name: `wrxml`
- copyright: (c) 2023 RENWare Software Systems
- author: Petre Iordanescu (petre.iordanescu@gmail.com)

Specifications:

- document cerinte initiale: `110-SRE-api_to_roefact_requirements.md` section `Componenta xl2roefact`
- INTRARI: fisier `f-JSON`
- IESIRI: fisier format XML conform cerintelor si sistemului ANAF E-Factura (cod: `f-XML`)

__init__

__main__

xl2roefact.main: Python package standard file to assure run as `python -m xl2roefact`.

Identification:

- code-name: `__main__`
- copyright: (c) 2023 RENWare Software Systems
- author: Petre Iordanescu (petre.iordanescu@gmail.com)

Deployments:

- Windows: MSI installer with EXE application.
- Linux: `xl2roefact` executable shell as wrapper for `xl2roefact.py`.

Specifications:

- command general format: `python -m xl2roefact [OPTIONS] COMMAND [ARGS]...`.
- help: `python -m xl2roefact --help`.

__version__

xl2roefact version info.

```
######## ####            ########  ######
#  ##  # #  #            #      # ##    ##
##    ## #  #            ###  ### #  ##  #
 ##  ##  #  #              #  #   #  ##  #
##    ## #  #              #  #   #  ##  #
#  ##  # #  #####          #  #   #  ##  #
#  ##  # #      #          #  #   ##    ##
######## ########          ####    ######

                         ######## ########  ######   ######  ########
#######   ######         #      # #     #  ##    ## ##    ## #      #
#     ## ##    ## ####### #  ##### #  ####  #  ##  # #  ##  # ###  ###
#  ##  # #  ##  # #     # #    #   #    #   #      # #  #####   #  #
#  ##### #  ##  # ####### #  ###   #  ###   #  ##  # #  #####   #  #
#  # #   #  ##  #         #  ##### #  #     #  ##  # #  ##  #   #  #
#  #  #  ##    ##         #      # #  #     #  ##  # ##    ##   #  #
####   #  ######          ######## ####    ########  ######    ####
```

Last update: February 29, 2024