

xl2roefact python library

- [xl2roefact python library](#)
 - [Library modules](#)
 - [Install library](#)
 - [Install from PyPi](#)
 - [Install from distribution packages](#)
 - [Aspecte tehnice referitoare la formatul fisierului JSON aferent facturii](#)
 - [API Reference](#)
 - [Download xl2roefact library](#)

Library modules

`xl2roefact` library (package) modules are:

- `rdinv` read an Excel file and extract invoice data to a JSON file format
- `wrxml` write, convert the JSON invoice file to a XML file format, respecting schemes required by *RO EFact* standard
- `chkxml` check generated XML file
- `ldxml` load an invoice (ie, its XML associated file) to *ANAF SPV system*
- `chkisld` check if an invoice is already loaded in *ANAF SPV system*
- `config_settings` define system settings & parameters mainly used in invoice info / data detection and extract from invoice Excel format file
- `app_cli` contains the code for `xl2roefact` application command line (CLI) format

Below is presented the ***skeleton logic*** of those modules which and where is relevant ie meaning where is not enough obvious from code or code complexity exceed usual limits (*for example more than 100 lines of code per function*). For more technical details and specification regarding modules [see API Reference](#)

Install library

Library can be installed using 2 methods:

- install from PyPi
- install from distribution packages

Install from PyPi

The library installation should be done using standard Python instruments:

```
pip install xl2roefact
```

Install from distribution packages

To install from distribution packages first download the package version intended to install ([see download section](#)), choose the package type (if you have no special option, then choose `WHEEL` format) and install it using `pip` as any other Python library installation (*detailed in Python official documentation*).

Aspecte tehnice referitoare la formatul fisierului JSON aferent facturii

Acest fisier este cel generat de catre aplicatie in urma executiei acesteia cu comanda `xl2json`. Structura de baza a acestui fisier este:

```
{
  "Invoice": {...},
  "meta_info": {...},
  "excel_original_data": {...}
}
```

Cheile de la primul nivel contin:

- **Invoice** - datele efective ale facturii
- **meta_info**
 - informatii referitoare la procesarea facturii si mapa de conversie a cheii `Invoice` din formatul `JSON` in formatul `XML` cerut de sistemul *RO E-Fact*
 - harta de ajutor in conversia formatului `JSON` in formatul `XML` acceptat de sistemul *RO E-Fact* (cheie `meta_info.map_JSONkeys_XMLtags`) si definitiile `XML` aferente (cheie `meta_info.invoice_XML_schemes`)
 - alte informatii despre fisierul Excel prelucrat (numele, worksheet cu factura, data si ora procesarii, CRC pentru verificare, etc)
- **excel_original_data** - informatiile originale din fisierul Excel, asa cum au fost ele identificate si gasite precum si locatia (adresele celulelor). Aceste informatii sunt utile in cazul in care exista neclaritati in urma procesului de conversie pentru "a intelege" de unde si cum arata informatiile originale din fisierul Excel

An [example of JSON generated file is available here](#)

API Refrence

Download xl2roefact library

- [Pachete instalare biblioteca Python formate WHEEL si DIST](#)

Last update: April 20, 2024