

- [xl2roefact](#)
 - [Facilitati \(Features\)](#)
 - [Instalarea](#)
 - [Instalarea aplicatiei xl2roefact](#)
 - [Instalarea bibliotecii Python \(package\) xl2roefact](#)
 - [Instalarea sablonului de factura emisa excel_invoice_template](#)
 - [Configurarea aplicatiei xl2roefact](#)
 - [Comenzile aplicatiei](#)
 - [about](#)
 - [settings](#)
 - [xl2json](#)
 - [Practici si reguli referitoare la continutul facturilor din Excel](#)
 - [Tutorial utilizare aplicatie](#)
 - [Organizarea informatiei](#)
 - [Exemplu de procesare a unei facturi](#)
 - [Formatul fisierului JSON](#)
 - [Documentatia tehnica](#)

xl2roefact

Available Versions

EXE**0.1.19****Python Script****0.1.20****Python package****0.1.20****Excel template****0.1.20****Documentation & support****Git Hub****0.1.20****Official Site****0.1**

Technical Doc

0.1.20

Facilitati (Features)

Aceasta aplicatie este "totul despre crearea de facturi electronice" din formatul Excel office (xlsx). Aplicatia poate genera factura in format JSON, XML, PDF si chiar o poate incarca in sistemul *RO E-Fact*.

(This application is all about electronic invoices creation from Excel office format (xlsx). It can generate invoice in JSON, XML, PDF formats and even upload it to *RO E-Fact* system.)

- Components
 - `x12roefact` **command line application** to create, manipulate and upload to RO E-Fact system Excel invoices
 - `x12roefact` **Python library** to use functions to create, manipulate and upload to RO E-Fact system Excel invoices
 - `excel_invoice_template` **an invoice template**
- Identification
 - code-name: `x12roefact`
 - copyright: (c) 2023 RENWare Software Systems
 - author: Petre Iordanescu (petre.iordanescu@gmail.com)
- Deployments
 - Windows: `x12roefact.exe` 64 bit CLI application (installable through a `MSI` package)
 - Linux: `x12roefact` executable CLI shell
- Specifications
 - command general format: `x12roefact [COMMAND] [OPTIONS]`
 - help: `x12roefact [COMMAND] --help`

Instalarea

Acest sistem consta din trei componente:

- o aplicatie de tip *linie de comanda / consola* ce permite realizarea operatiilor necesare de procesare a facturilor in Excel (`x12roefact` CLI application)
- o biblioteca (`x12roefact`) ce permite utilizarea functionalitatilor de procesare a facturilor Excel in mod programatic in alte sisteme. Biblioteca este realizata in Python 3 iar instalarea ei se poate face cu instrumentele

standard Python

- un sablon de factura emisa (`excel_invoice_template`) ce poate fi utilizat "ca atare" pentru emitere facturi

Instalarea aplicatiei xl2roefact

Pachetele de instalare se gasesc in directorul `dist/` ca arhive `ZIP` . Pachetele disponibile contin in numele lor versiunea de aplicatie utilizata si versiunea sistemului de operare pentru care sunt disponibile. (EN: *Installation package is found in `dist/` directory in archive (zip) files. Available packages are identified by released versions and operating systems.*)

Pachetele contin un script de instalare sub forma standard `MSI` pentru Windows si `DEB` pentru Linux Debian (verificati disponibilitatea pentru sistemul de operare folosit de dvs). (EN: *Packages contains an installation script as `MSI` for Windows and `DEB` for Debian Linux based systems (check availability for your operating system).*)

Instalarea bibliotecii Python (package) xl2roefact

Instalarea acestuia se face cu instrumentele standard Python. Recomandarea este pentru instalarea simpla cu: `pip install xl2roefact` , biblioteca fiind disponibila in repository-ul standard `PyPy`. Pentru instalarea din surse, biblioteca poate fi descarcata din [GitHub](#).

Instalarea sablonului de factura emisa excel_invoice_template

Acest "pachet" este un director (ce nu necesita instalare speciala ci simpla copiere locala acolo unde va fi utilizat). El contine: - fisierul pentru factura `invoice_template_CU_tva.xlsx` - un document descriptiv cu "reguli" recomandate in utilizarea acestui sablon [README_excel_invoice_rules.md](#)

Configurarea aplicatiei xl2roefact

Parametrii de configurare a aplicatiei se gasesc in fisierul `config_settings.py` . Acestia sunt sub elaborati in limbaj Python prin utilizarea conventiilor de constante conform recomandarilor PEP (numele capitalizat) si sunt acompaniati de linii de explicatii privind aplicabilitatea lor. (EN: *Configuration parameters are placed in file `config_settings.py` . These are in Python form presented using constants PEP recommendations (all upper case) and accompanied by some help lines to understand and maintain them.*)

Comenzile aplicatiei

Interfata aplicatie este realizata utilizind conventiile si practicile uzuale pentru aplicatii tip linie de comanda consola. Pentru informatii privind comenzile se poate folosi optiunea de **help**, disponibila atat la nivelul general: `xl2roefact --help` , cit si la nivel detaliat pentru fiecare comanda: `xl2roefact [COMANDA] --help` .

about

Afiseaza informatii despre aceasta aplicatie (copyright, scop, etc).

settings

Afiseaza parametrii de configurare a aplicatiei. [Vezi sectiunea de configurare](#)

xl2json

Transforma fisierul (fisierele) Excel in forma JSON pentru utilizare ulterioara ca forma de date standardizat pentru schimbul de informatii cu alte sisteme electronice. Formatul JSON utilizat contine:

- informatiile aferente facturii (cheie: `Invoice`)
- o 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 (alte chei din `meta_info`)
- datele preluate din formatul original Excel (cheie `excel_original_data`) - acestea sunt utile pentru depanare in caz ca aceasta este necesara in cazul specific al fisierului Excel folosit de dvs

Practici si regului referitoare la continutul facturilor din Excel

Acest capitol se refera la modul in care este "tratata" continutul fisierului Excel cu factura, mai exact la modalitatea in care informatia facturii este cautata, identificata si gasita in scopul de a fi salvata in oricare din formatele de "factura electronica / E-Fact".

Utilizarea sablonului de factura Excel ce este livrat impreuna cu aplicatia **ESTE O VARIANTA DE LUCRU RECOMANDATA**, dar nu obligatorie. Chiar si in cazul utilizarii acestuia, prin modificarea "structurii" acestuia, informatia poate ajunge *nerecognoscibila / neidentificabila* total sau partial daca nu sunt urmate regulile expuse.

In general trebuie facuta diferenta intre datele facturii si modul in care aceasta va fi tiparita (va aparea la tiparire / previzualizare).

Mai exact **continutul informational** al facturii nu trebuie nici confundat si nici mixat cu **formatul de afisare al acesteia** (layout). Pentru acesta din urma se recomanda a fi folosite cu precadere *regulile de formatare* din Excel si nu cele stocare a datelor. Un exemplu este un numar zecimal oarecare unde:

- una este valoarea introdusa intr-o celula (de ex cu 3 zecimale) si
- alta este valoarea afisata (cu 2 zecimale) - aceasta din urma trebuie obtinuta prin formatarea celulei respective de a afisa 2 zecimale prin rotunjire insa valoarea efectiva trebuie sa fie cea originala cu 3 zecimale, lucru (diferenta) care se poate vedea la editarea continutului celulei.

-#TODO ...tbd (se va lua din `_docstring` aferent `config.settings.py`)

Tutorial utilizare aplicatie

Organizarea informatiei

Aplicatia `xl2roefact` "promoveaza" structurarea informatiei procesate astfel incit sa fie evitata situatia "*de aglomerare*" a directorului curent cu fisiere ce trebuiesc identificate si izolate in situatia in care se fac *procesari in masa* (pe mai multe fisiere / facturi sursa).

Astfel, aplicatia se asteapta ca fisierele Excel sursa (*adica facturile de procesat*) sa fie copiate in directorul `invoice_files/` de unde vor fi citite si tot aici vor fi create fisierele rezultate (JSON, XML, etc). Acest director este relativ la directorul curent de unde este lansata aplicatia si considerat "*implicit*" cu acest nume dar daca se doreste un alt director acest lucru poate fi facut folosind parametrul `--files-directory` (sau prescurtat `-d`) la lansarea aplicatiei astfel:

```
xl2roefact -d "calea si numele directorului dorit"
```



Nota

Ghilimelele sunt necesare numai daca numele si calea (path) contin caracterul spatiu.

Exemple:

- pentru stabilirea directorului curent ca sursa pentru fisierele factura Excel: `xl2roefact -d ./`
- procesarea tuturor facturilor din luna *iunie*, copiate intr-un director dedicat sub directorul curent:
`xl2roefact -d ./facturi_iunie/`

Exemplu de procesare a unei facturi

- se creaza directorul `invoice_files`
- se copiaza factura `factura_A.xlsx` in acest director apoi se revine in directorul anterior daca acesta a fost schimbat pentru efectuarea copierii
- se lanseaza aplicatia: `xl2roefact xl2json`

In urma acestor operatii, in directorul `invoice_files` vor rezulta:

```
invoice_files/
├─ factura_A.xlsx # fisierul Excel original
└─ factura_A.json # fisierul JSON rezultat in urma procesarii
```

- `factura_A.xlsx` ca fiind fisierul Excel original cu factura
- `factura_A.json` acesta fiind fisierul format JSON rezultat in urma procesarii si ce poate fi folosit pentru interschimbarea electronica a informatiei intre sisteme

Formatul fisierului JSON

Structura de baza a fisierului JSON aferent unei facturi este:

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

Cheile de la primul nivel reprezinta:

- **Invoice** - #TODO tbd...
- **meta_info** - #TODO tbd...
- **excel_original_data** - #TODO tbd...

Documentatia tehnica

Last update: January 23, 2024