

RENware Software Systems

xl2roefact



version 0.9.dev0

Legaturi externe utile:

- [Web Site](#). (Pentru acces corect la toate referintele din acest document vizitati site-ul dedicat acestui sistem.)
- [Pachet PyPi](#)
- [Surse GitHub](#)
- [Referinta dezvoltare software - biblioteca Python](#)
- [Referinta API](#)

Facilitati



MSI installer YES



standalone EXE YES

script Python3

YES

pachet DEB

in curind

format JSON

YES

format XML

YES

format PDF

YES

format RO eFact

YES

Aceasta componenta este "totul despre crearea de facturi electronice" din formatul Excel office (xlsx). Aplicatia poate genera factura in format JSON, XML, PDF si o poate incarca in sistemul RO E-Fact¹.

Aceasta componenta ofera urmatoarele facilitati (acestea fiind obiectivele fundamentale ale componentei):

- **transformarea facturilor din Excel in formatul XML** cerut de catre sistemul ANAF RO E-Fact pentru incarcare

- **incarcarea acestora** in sistemul ANAF RO E-Fact¹
- **transformarea facturilor din Excel intr-un format JSON** intermediar, independent de platforma si care permite integrarea acestora cu alte sisteme (standard *REST*)
- **generarea facturii in format PDF** pentru transmiterea acesteia catre client, semnarea electronica, tiparirea si arhivarea acesteia in format fizic (in general manipularea facturii in format "*human readable*")

Componenta ofera doua instrumente pentru realizarea si indeplinirea acestor obiective:

- `xl2roefact` o **aplicatie de tip linie de comanda** (disponibila pentru sistemele de operare Windows, Linux si MacOS)
- `xl2roefact` PyPi **biblioteca standard Python** utilizabila pentru dezvoltari proprii in scopul extinderii altor sisteme existente (*custom development*)

Instalarea aplicatiei xl2roefact

Instalarea aplicatiei *xl2roefact* este disponibila in urmatoarele variante:

- pentru **Windows**:
 - **MSI** pachet instalare pentru *Windows*
 - **EXE** executabil *Windows in format "portabil" (un singur fisier)*
- pentru **Linux**:
 - *...in curind...* **DEB** pachet instalare pentru *Linux Debian*
 - *...in curind...* **RPM** pachet instalare pentru *Linux*
 - *...in curind...* **APPIMG** executabil *Linux in format "portabil" (un singur fisier)*
- pentru **Mac OS X**
 - *...in curind...* **DMG** pachet instalare pentru *MacOS* <!--
- ca **script Python** indiferent de sistemul de operare;
 - **Pachet Python** biblioteca / libraria completa pe PyPi (inclusiv sursele)

Pentru acces la pachetele de instalare vezi [sectiunea de descarcare resurse](#).

Note: * *utilizarea ca script Python necesita existenta ca mediul Python3 min 3.10 sa fie instalat local* * numele pachetelor includ versiunea de aplicatie utilizata si sistemul de operare pentru care sunt disponibile * pentru echivalent utilizare *portabila pentru Linux* se poate instala biblioteca Python dupa care devine utilizabil scriptul Python "ca orice alta comanda Linux"

Configurarea aplicatiei xl2roefact

Parametrii de configurare aplicatiei se gasesc in fisierul `config_settings.py`. Acestia sunt sub elaborati in limbaj Python prin utilizarea conventiilor de constante conform recomandarilor PEP (numele capitlatizat) si sunt acompaniti de linii de explicatii privind aplicabilitatea lor.

Configurare aplicatiei se poate face interactiv si din aplicatie. Pentru a obtine help referitor la detaliile comenzi se va folosi

```
xl2roefact settings --help
```

Configurarile existente si regulile recomandate in configurarea aplicatiei se afiseaza folosind comanda:

```
xl2roefact settings --rules
```

Configurarea din fisier extern

Configurarea aplicatiei se poate face si prin intermediul unui fisier extern numit "*sablon de configurare*" (*en: configuration template*). Sablonul permite configurarea aplicatiei prin modificarea fragmentelor de text care trebuiesc cautate in fisierul Excel pentru identificarea diverselor informatii aferente facturii.

Sablonul este in format [YAML](#) iar informatiile ce trebuiesc descrise sunt explicate individual in comentarii insotitoare. De asemenea este util a fi citite si recomandarile date in pagina de descriere a aplicatiei.

Pentru a beneficia de cobfigurare facute de dumneavoastra trebuie sa creati un fisier `app_settings.yml` in directorul curent din care lansati aplicatia, fisier ce contine noile configurari dorite. **Numele fisierului este obligatoriu a fi respectat.**



Fisiere de configurare multiple

De retinut ca acest fisier este considerat (daca exista) cel din directorul curent de unde lansati aplicatia. Deci daca v-ati creat mai multe directoare de lucru (de exemplu pentru clienti diferiti) puteti crea fisiere de configurare specifice, cite unul in fiecare director.

Fisier de configurare global

În condițiile folosirii kitului MSI pentru o instalare locală a aplicației (cu utilizări multiple și repetate) și în situația în care se dorește schimbarea configurării implicite a aplicației se vor urma acești pași:

- în directorul de instalare a aplicației se va crea dacă nu există directorul `data/`
- în acest director se va crea un fișier `app_settings.yml` cu configurarea globală dorită

Această configurare înlocuiește configurarea implicită și se va aplica global în utilizarea aplicației. În continuare configurările existente în directorul curent *suprașcriu configurarea globală* (se aplică cu precedentă).

Aici puteți găsi pentru descărcare un model de șablon de configurare.

Utilizare nomenclator de furnizori

Aplicația *x/2roefact* permite utilizarea datelor pentru furnizori din fișiere externe (în locul informațiilor din fișierele Excel) lucru ce poate fi folositor în următoarele situații:

- când utilizatorul aplicației o face în scopuri personale și multe facturi emise îl au *pe el ca furnizor*. Acest lucru permite ca informația din Excel referitoare la furnizor să fie sumară sau să lipsească, factura finală format `PDF` fiind generată cu aplicația
- când utilizatorul aplicației o folosește pentru a emite facturi pentru alte firme și astfel este mai comod să folosească fișiere cu datele acestor firme decât să introducă informația în fiecare factură
- când se dorește ca datele furnizorului să fie preluate dintr-un sistem extern ce le poate exporta ca și fișiere

Reguli generale de utilizare

Această secțiune descrie regulile generale ce trebuie să aibă în vedere pentru o completă și corectă utilizare a facilității "Nomenclator furnizori":

- Nomenclatorul de furnizori se va completa într-unul sau mai multe fișiere de date (de tip text, vezi mai jos formatul exact).
- Un fișier acomodează un singur furnizor. Pentru mai mulți furnizori se vor folosi fișiere diferite.
- Numele fișierului (fără extensie) trebuie să coincidă cu o cheie alternativă a furnizorului respectiv. Prin cheie alternativă se înțelege acea cheie care este unică și poate asigura regăsirea furnizorului prin folosirea ei. Ca și exemple din practicile curente ar fi cimpul numit uzual `code` sau `code_name` existent în mai toate sistemele de business. Acesta are avantajul unicității și a unei reprezentări "umane" (*en: human readable*). Desigur un cimp de tip cheie primară / ID este ideal dar de obicei acesta este tehnic iar valoarea să nu ofere prea multe indicații.
- Formatul fișierului este `YAML` standard, fără folosirea de modele de date complexe, astfel încât o eventuală conversie `JSON <--> YAML` să poată fi realizată manual în ambele sensuri și fără necesitatea unor cunoștințe avansate ci la nivel de redefinire a numelor cheilor.

Locatia nomenclatorului

Fisierele cu datele furnizorilor pot sta in urmatoarele locatii:

- *directorul curent* este locatia cu prioritatea maxima si in caz de "duplicate" ale unui fisier, cel de aici va fi luat in considerare
- *directorul* `data/` din locatia unde este instalata aplicatia

Recomandari si practici uzuale:

- In situatiile in care sistemul este instalat pe un computer ce se foloseste frecvent cu aplicatia *xl2roefact* si exista un set de furnizori frecvent folositi se recomanda folosirea directorului `data/` pentru stocarea fisierelor nomenclator astfel incit sa poata fi refolosite usor.
- In situatia folosirii a "multe" fisiere date furnizori se recomanda crearea unui director dedicat in locatia utilizata (vezi mai sus) si acesta sa fie referit in numele fisierului.

Utilizarea nomenclatorului

Pentru a folosi cu aplicatia un fisier tip nomenclator furnizor se va utiliza optiunea:

```
xl2roefact xl2json -o fisier_furnizor
```

unde `fisier_furnizor` este numele fisierului ce contine datele unui furnizor. Locatia acestui fisier este relativa la [locatia considerata pentru folosire](#)

Sablon pentru nomenclator de furnizori

Sablonul este proiectat pentru utilizarea in facturile emise si contine numai informatiile necesare in acest scop.

Astfel cimpurile existente trebuiesc pastrate, adica nu vor fi sterse.

Vor fi *respectate si completate* corespunzator cimpurile specificate ca *obligatorii* (*en: mandatory* in comentariile aferente fiecarui cimp).

Pentru acele cimpuri pentru care informatia este necunoscuta sau considerata irelevanta se va completa cu `null`.

Se vor putea adauga orice alte cimpuri suplimentare cu conditia sa fie respectat formatul fisierului (YAML). acestea nu vor fi folosite de catre aplicatie, ci pur si simplu ignorate.

[Aici puteti gasiti un model de sablon de configurare.](#)

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**, dispobilia atit la

nivelul general:

```
xl2roefact --help
```

cit si la nivel detaliat pentru fiecare comanda

```
xl2roefact [COMMAND] --help
```

Lista comenzilor:

- **about** - Afiseaza informatii despre aceasta aplicatie (copyright, scop, etc)
- **settings** _ Afiseaza parametrii de configurare a aplicatiei. [Vezi sectiunea de configurare a aplicatiei](#)
- **xl2json** - Transforma fisierul (fisierele) Excel in forma JSON pentru utilizare ulterioara ca forma de date standardizat pentru schimbul de informatii cu alte sisteme electronice

Comenzile detaliate:

xl2roefact

Application global information (command agnostic).

Usage:

```
xl2roefact [OPTIONS] COMMAND [ARGS]...
```

Options:

<code>-V, --version</code>	show application version
<code>--install-completion</code>	Install completion for the current shell.
<code>--show-completion</code>	Show completion for the current shell, to copy it or customize the installation.
<code>--help</code>	Show this message and

about

Provide a short application description.

Usage:

```
xl2roefact about [OPTIONS]
```

settings

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

Args:

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

Usage:

```
xl2roefact settings [OPTIONS]
```

Options:

```
-r, --rules show settings recommended update  
rules
```

xl2json

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

Args:

```
`invoice_type_code`: invoice type (for exaple regular invoice or storno) as this info is not  
usually subject of Excel file. Default to `380` (regular / usual invoice)  
`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  
`owner_datafile`: File to read invoice supplier (owner) data instead Excel.  
`verbose`: show detailed processing messages". Defaults to `False`.
```

Usage:

```
xl2roefact xl2json [OPTIONS] [FILE_NAME]
```

Options:


```

--invoice-type [380]          [default: 380]
[FILE_NAME]                  files to process
                              (wildcards
                              allowed)
                              [default:
                              *.xlsx]
-d, --files-directory DIRECTORY
                              directory to be
                              used to look for
                              Excel files (if
                              default
                              directory does
                              not exists will
                              consider current
                              directory
                              instead).
                              [default:
                              invoice_files/]
-o, --owner-datafile FILE     File to read
                              invoice supplier
                              (owner) data
                              instead Excel.
-v, --verbose                 show detailed
                              processing
                              messages
--help                        Show this
                              message and

```

Practici si reguli 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.

Reguli recomandate in configurarea aplicatiei pe specificul Excel al facturilor dumneavoastra

Reguli recomandate pentru adaptarea aplicatiei la modelul dvs de factura in Excel:

- 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 liste (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 celului Excel a se utiliza formatul standard de data (date) si modificati formatul de afisare in formatul dorit pe factura tiparibila

(en-us) Recommended rules to configure the application to your Excel invoice model:

- 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 sqaued 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

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 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 recomandat pentru stocarea facturilor in Excel:

```
md 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

Aspecte tehnice referitoare la formatul fisierului JSON aferent facturii

Acest fisier este cel generat de catre aplicatie in urma executiei acesteia cu comanda `xl2json`. Formatul JSON are urmatoarra structura de baza:

```
{
  "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

Detalii suplimentare despre formatul JSON se gasesc in documentația [Referinta dezvoltare software](#).

Referinta dezvoltare software

Documentatia "[Referinta dezvoltare software](#)" ofera detail necesare pentru utilizarea bibliotecii sursa in dezvoltari specifice, extindere si integrare cu alte sisteme.

Date identificare

- part number (p/n): `0000-0095-xl2roefact`
- producator si copyright: RENWare Software Systems (referinte detalii tehnice: Petre Iordanescu, petre.iordanescu@gmail.com)

License

Note

1. Toate interactiunile cu sistemul *ANAF RO E-Fact* necesita o *conexiune la internet* si un set de *credentiale ANAF RO E-Fact ale companiei* pentru care se incarca factura. In lipsa acestora, fisierul `XML` generat de aplicatie poate fi incarcat ulterior (de ex de catre departmentul contabilitate) [↩](#)[↩](#)
-

Last update: May 14, 2024