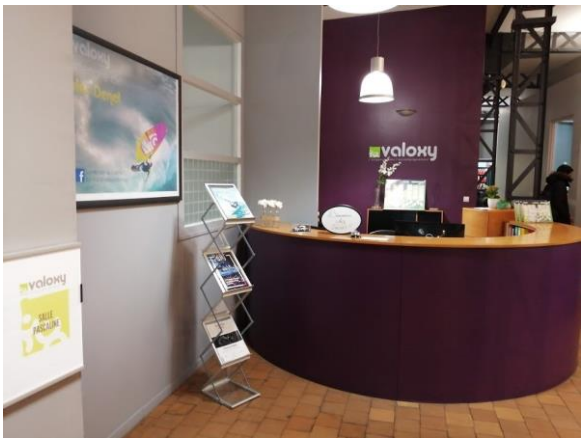
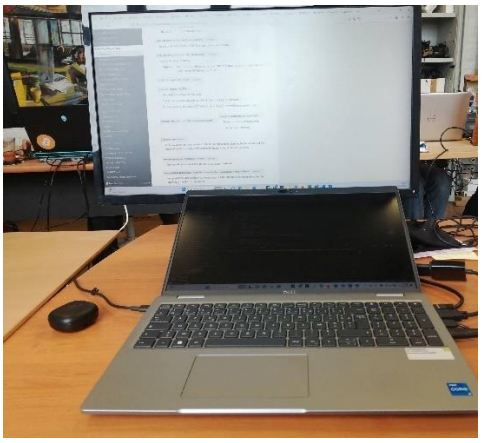
Stage Valoxy 2024

Introduction :

a) Valoxy

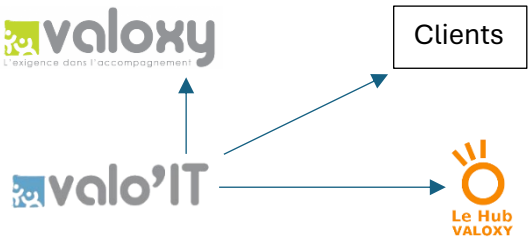




Valoxy est un cabinet d'expertise comptable fondé en 2004 par Ludovic Tiberghien à Lille et Saint-Omer avec une devise forte : « L'exigence dans l'accompagnement ». Toute entreprise doit tenir une comptabilité complète et détaillée. Valoxy propose donc ses services aux entreprises qui souhaitent externaliser leur comptabilité. La société offre des services adaptés à chaque entreprise. Aujourd'hui, Valoxy dispose d'un peu plus de 700 clients qui proviennent de secteurs divers et variés. Le groupe Valoxy se concentre sur ses clients au travers d'un réseau de proximité réparti sur dix sites dans les Hauts de France. Le cabinet s'est doté d'une équipe d'experts spécialisés comptant environ 170 personnes. https://valoxy.org/

b) Valo'it

Valo'IT est une société de services informatiques faisant partie du groupe Valoxy qui se situent au siège social de l'entreprise Valoxy à La Madeleine. Ses services englobent la mise en place de solutions IT, une offre téléphonique, la création de sites internet et la simplification digitale. L'équipe est restreinte, composée actuellement de seulement 4 employés. Le HUB Valoxy (espace de coworking) est administré par Valo'IT. Valo'IT compte Valoxy comme principal client, mais également d'autres entreprises qui sont également clientes de Valoxy. Étant rattaché(e) à la direction du service informatique. https://valoit.fr/



c) Contexte

Chorus Pro est une plateforme gérer pas l'AIFE (L'agence pour l'informatique financière de l'état) créer en 2017. Initialement, la plateforme concerne les entreprises qui sont fournisseurs à l'état ou les entreprises crée, édite et transmet des factures directement aux services de l'état.

L'article 26 du 16 aout 2022 de la loi des finances impose la généralisation de la facture électronique entre entreprise et la transmission de ces factures à l'administration fiscale.

Cette loi a pour but de réduire la fraude de TVA. Il était initialement prévu en 2023, mais retardé pour 2025. En 2025, l'utilisation de la plateforme sera obligatoire pour les grandes entreprises, en 2026 pour les entreprises de taille intermédiaires et en 2027 pour les très petit/ petites et moyennes entreprises.

https://communaute.chorus-pro.gouv.fr/sinformer-sur-la-facturation-electronique-b2b/

https://communaute.chorus-pro.gouv.fr/documentation/facturation-electronique-interentreprises/

https://communaute.chorus-pro.gouv.fr/category/actualite/facturation-electronique/

Chorus Pro à aussi créer un portail de qualification. Un site indépendant de Chorus Pro qui permet aux utilisateurs de tester la plateforme en utilisant des données fictives.

https://communaute.chorus-pro.gouv.fr/portail-de-qualification/

Anis de Flavigny est un client de Valoxy qui demande une solution qui permet d'envoyer des factures facilement à Chorus Pro. Cette tache m'a été demander.

d) Les technologies utilisées

Initialement, on a voulu utiliser javascript car l'équipe était plus familière avec ce langage. Mais avec javascript, je rencontrais des difficultés à effectuer la première tâche. Donc j'ai recherché d'autres manières possibles. Je l'avais réussi à le faire sur python. Donc j'ai décidé de faire tout le reste du projet sur python.

Le SGBD utilisé par Valoxy est Microsoft SQL Server.

1) Convertir un PDF en Factur-X

Factur-X (ZUGFeRD 2.2 en Allemagne) est un standard franco-allemand pour les factures électroniques qui utilise un format mixte : PDF pour les utilisateurs et données XML pour le traitement automatique. Il est basé sur la Norme Sémantique Européenne EN 16931, établie par la Commission Européenne en octobre 2017 pour harmoniser les échanges de factures électroniques dans l'Union européenne.

https://fnfe-mpe.org/factur-x/

Le standard Factur-X est un PDF/A-3b avec un fichier XML en pièce jointe.

PDF/A-3b est un standard ISO qui autorise la jointure d'autre fichiers dans le PDF.

Prérequis :

- Le logiciel Ghostscript :

https://ghostscript.com/

- Les Library python suivante :

```
pip install logging

pip install dateTime
```

- Le fichier pdf à convertir
- Le fichier xml à mettre en pièce jointe
- Un fichier profil icc
- Un fichier postscript

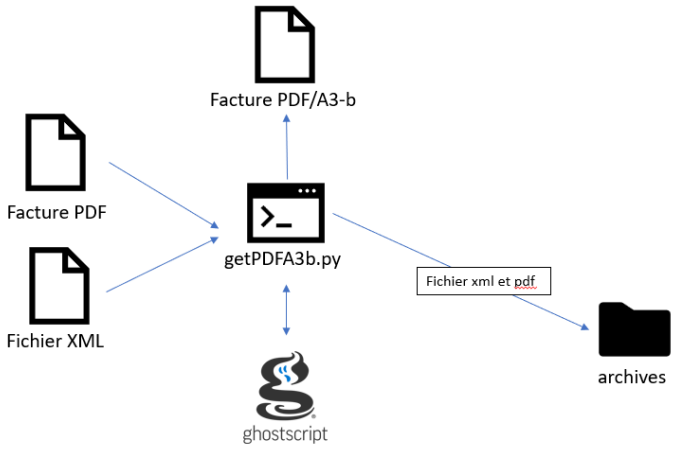
Ghostscript est une suite logicielle permettant le traitement des formats de fichiers PostScript et PDF.

Un profil ICC est un fichier numérique d'un format particulier décrivant la manière dont un périphérique informatique restitue les couleurs. Ceci est nécessaire pour la conversion de pdf en pdfA/3-b

https://www.color.org/profiles2.xalter

Le fichier postscript va permettre d'attacher les fichiers icc et xml. On peut récupérer ce fichier dans ce path:

"C:\Program Files\gs\gs10.02.1\lib\zugferd.ps"



Ce programme permet d'exécuter une ligne de commande dans le logiciel Ghostscript :

```
# Fonction pour convertir un fichier PDF en PDF/A avec Ghostscript
def convert_to_pdfa(input_path, output_path, ps_path, xml_path, icc_path):
    # Commande pour appeler Ghostscript avec les options appropriées pour la conversion
PDF/A
    command = [
        gs_path,
        f'--permit-file-read="{folder_path}"',  # Autoriser la lecture de fichiers dans le
dossier spécifié
        "-sDEVICE=pdfwrite",
        "-dNOSAFER",
        "-dPDFA=3",
        "-sColorConversionStrategy=UseDeviceIndependentColor",
        f"-sZUGFeRDXMLFile={xml_path}",  # Chemin vers le fichier XML ZUGFeRD
        f"-sZUGFeRDProfile={icc_path}",  # Chemin vers le profil ICC
        "-sZUGFeRDVersion=2p2",
        "-sZUGFeRDConformanceLevel=EN 16931",
        f"-o {output_path}",  # Chemin de sortie du fichier PDF/A
        os.path.abspath(ps_path),  # Chemin absolu du fichier PostScript
        os.path.abspath(input_path)  # Chemin absolu du fichier PDF d'entrée
    ]

    try:
        # Exécuter la commande Ghostscript pour convertir le fichier PDF
        subprocess.run(command, check=True)
        return True
    except subprocess.CalledProcessError as e:
        # En cas d'erreur, lever une exception ConversionError
        error_message = f"Error during PDF conversion: {e}"
        destination_path = os.path.join(errorPath, os.path.basename(input_path))
        shutil.move(input_path, destination_path)
        os.remove(xml_path)
        os.remove(output_path)
        print("err " +input_path +"\n" + error_message)
        logErr.error(input_path + "\n" + error_message)
        log.error(input_path + "\n" + error_message)
```

```
        return False
```

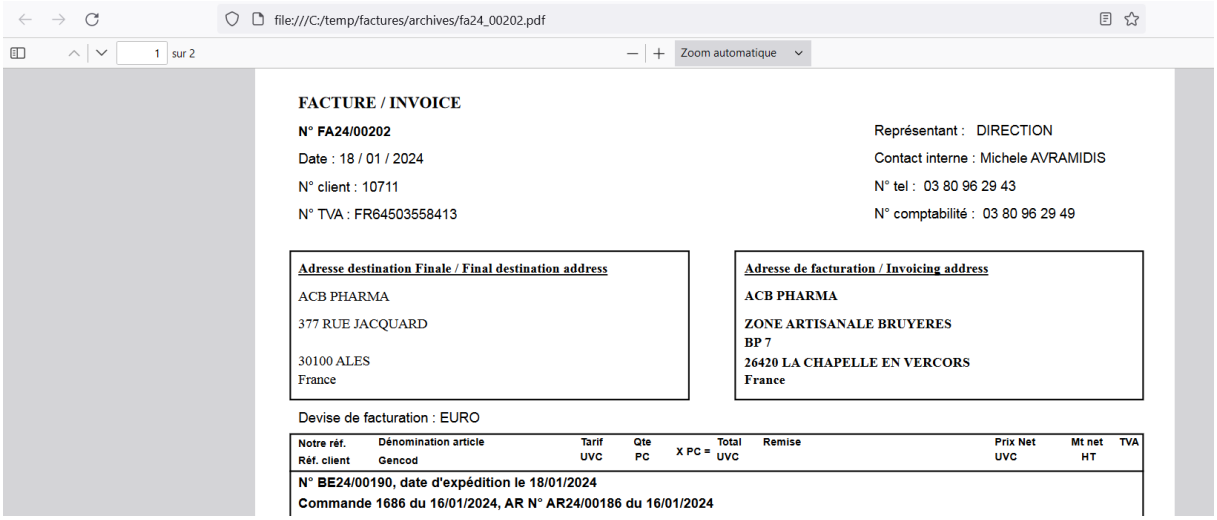
Dans le tableau de la variable `command` on a une série de commandes ghostscript.

gs_path	Va permettre d'executer la commande
--permit-file-read="{path}"	Permet au programme de lire les fichier
"-sDEVICE=pdfwrite"	On indique à Ghostscript d'utiliser le pilote d'écriture PDF
"-dNOSAFER"	Permet de donner les droits au programme de modifier, lire ou supprimer des fichiers
"-sColorConversionStrategy=UseDeviceIndependentColor"	La conversion des couleurs est basée sur des normes spécifiques plutôt que sur les caractéristiques du périphérique
"-sZUGFeRDXMLFile={xml_path}"	Chemin d'access pour lire le fichier xml
"-sZUGFeRDProfile={icc_path}"	Chemin d'access pour lire le fichier icc
"-sZUGFeRDVersion=2p2"	Défini la version de zugferd (zugferd 2.2 = factur-x)
"-sZUGFeRDConformanceLevel=EN 16931"	Défini la norme
"-o {output_path}"	Chemin pour le fichier converti
os.path.abspath(ps_path)	Chemin d'access pour lire le fichier postscript
os.path.abspath(input_path)	Chemin d'access pour lire le fichier à convertir

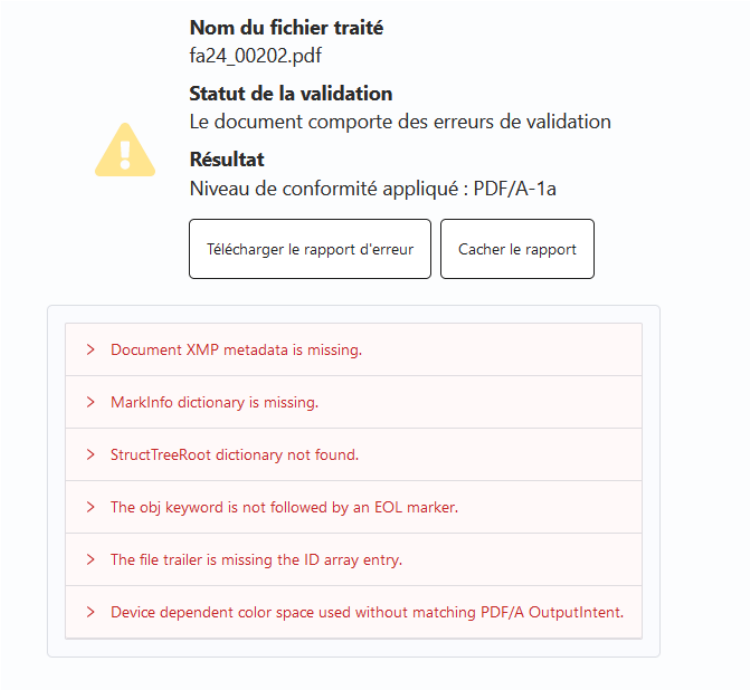
https://stackoverflow.com/questions/77840665/xml-file-doesnt-get-attached-into-the-pdf-while-using-ghostscript

Test :

Voici une facture :

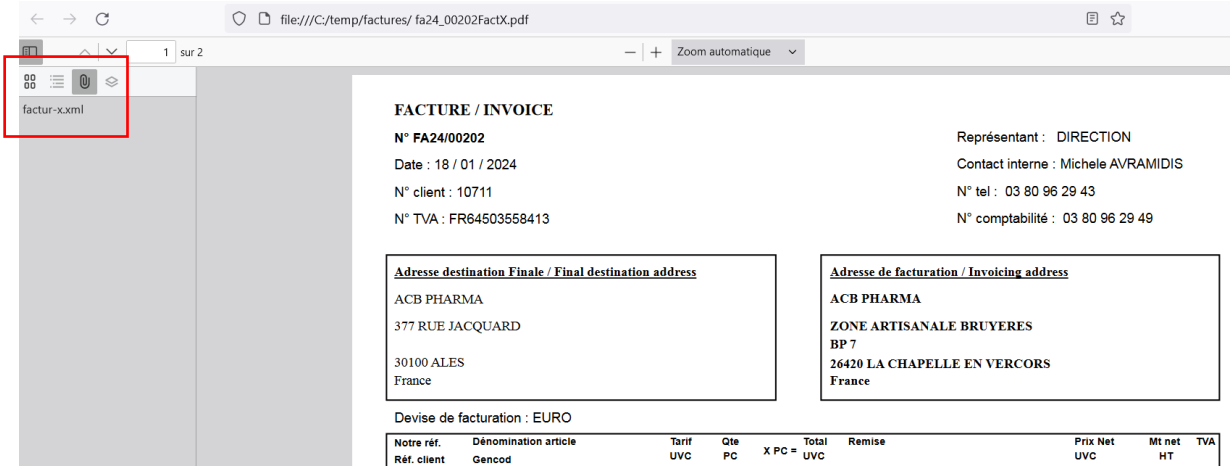


En utilisant ce site :
https://avepdf.com/fr/pdfa-validation



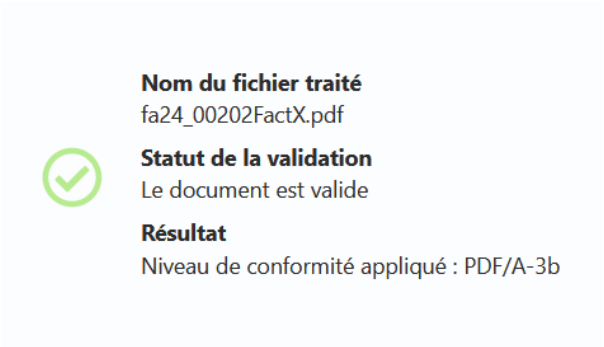
Le fichier n'est donc pas conforme au format factur-x

On exécute le programme, on obtient ce fichier :



On a tout d'abord un fichier xml en pièce jointe.

Ensuite il est conforme au standard PDF/A-3b



2) Créer un fichier XML

Pour qu'un document soit conforme au format factur-x (conforme aussi à la norme EN16931), il faut que le fichier XML en pièce jointe soit conforme au CII (Cross Industry Invoice)

Le CII est une norme internationale créer par l'UNECE (Commission économique pour l'Europe des Nations unies) pour les factures, rendant la facturation plus facile entre différentes entreprises. Il simplifie les processus, réduit les erreurs et accélère les paiements. En normalisant les données, il améliore la visibilité financière et aide les entreprises à être plus efficaces, économiser des ressources et respecter les règlements.

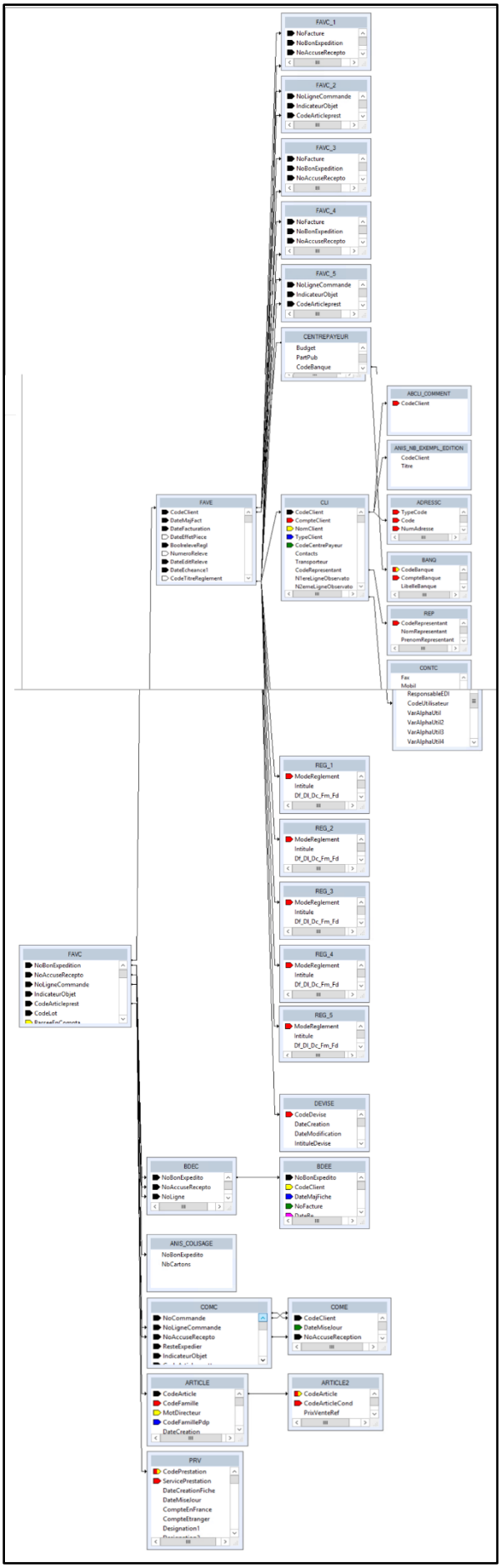
https://unece.org/trade/uncefact/e-invoice

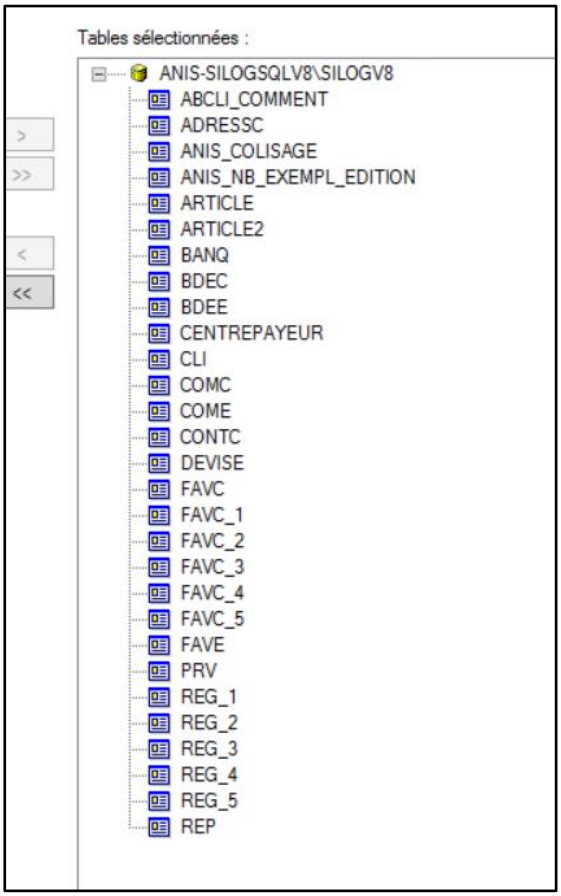
Prérequis :

- Microsoft SQL server management studio
- Installé les Library python suivant :

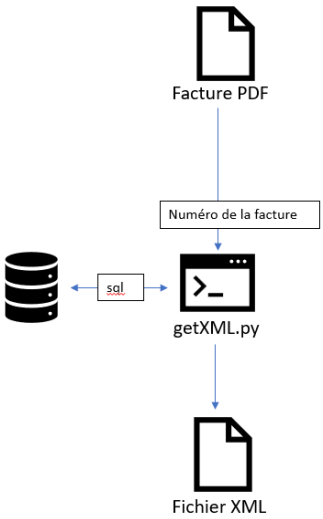
```
pip install pyodbc
pip install PyPDF2
pip install logging
pip install dateTime
```

On a une base de données contenant les données des factures :


FAVC_1
NoFacture
NoBonExpedition
NoAccuseRecepto
FAVC_2
NoLigneCommande
IndicateurObjet
CodeArticleprest
FAVC_3
NoFacture
NoBonExpedition
NoAccuseRecepto
FAVC_4
NoFacture
NoBonExpedition
NoAccuseRecepto
FAVC_5
NoLigneCommande
IndicateurObjet
CodeArticleprest
CENTREPAYEUR
Budget
PartPub
CodeBanque
ABCLI_COMMENT
CodeClient
ANIS_NB_EXEMPL_EDITION
CodeClient
Titre
FAVE
CodeClient
DateAdqFact
DateFacturation
DateEffetPiece
BoolreleveRegl
NumeroReleve
DateEditReleve
DateEcheance1
CodeTitreReglement
CLI
CodeClient
CompteClient
NomClient
TypeClient
CodeCentrePayeur
Contacts
Transporteur
CodeRepresentant
NYemeLigneObservato
NZemeLigneObservato
ADRESSC
TypeCode
Code
NumAdresse
BANQ
CodeBanque
CompteBanque
LibelleBanque
REP
CodeRepresentant
NomRepresentant
PrenomRepresentant
CONTC
Fax
Mobil
ResponsableEDI
CodeUtilisateur
VarAlphaUtil
VarAlphaUtil2
VarAlphaUtil3
VarAlphaUtil4
REG_1
ModeReglement
Intitule
Df_Dl_Dc_Fm_Fd
REG_2
ModeReglement
Intitule
Df_Dl_Dc_Fm_Fd
REG_3
ModeReglement
Intitule
Df_Dl_Dc_Fm_Fd
REG_4
ModeReglement
Intitule
Df_Dl_Dc_Fm_Fd
REG_5
ModeReglement
Intitule
Df_Dl_Dc_Fm_Fd
FAVC
NoBonExpedition
NoAccuseRecepto
NoLigneCommande
IndicateurObjet
CodeArticleprest
CodeLot
DEVISE
CodeDevise
DateCreation
DateModification
IntituleDevise
BDEC
NoBonExpedito
NoAccuseRecepto
NoLigne
BDEE
NoBonExpedito
CodeClient
DateMajFiche
NoFacture
ANIS_COLISAGE
NoBonExpedito
NbCartons
COMC
NoCommande
NoLigneCommande
NoAccuseRecepto
ResteExpedier
IndicateurObjet
COME
CodeClient
DateMiseJour
NoAccuseReception
ARTICLE
CodeArticle
CodeFamille
MotDirecteur
CodeFamillePdp
DateCreation
ARTICLE2
CodeArticle
CodeArticleCond
PrixVenteRef
PRV
CodePrestation
ServicePrestation
DateCreationFiche
DateMiseJour
CompteEnFrance
CompteEtranger
Designation1


Tables sélectionnées :
ANIS-SILOGSQLV8\SILOGV8
ABCLI_COMMENT
ADRESSC
ANIS_COLISAGE
ANIS_NB_EXEMPL_EDITION
ARTICLE
ARTICLE2
BANQ
BDEC
BDEE
CENTREPAYEUR
CLI
COMC
COME
CONTC
DEVISE
FAVC
FAVC_1
FAVC_2
FAVC_3
FAVC_4
FAVC_5
FAVE
PRV
REG_1
REG_2
REG_3
REG_4
REG_5
REP

Tout d'abord, j'ai commencé par examiner le fichier XML pour répertorier les données importantes, ensuite je cherche ces données dans la base de données, puis je crée un programme qui place les données dans un modèle de fichier XML conforme au format Factur-X, et enfin le programme génère le fichier XML.



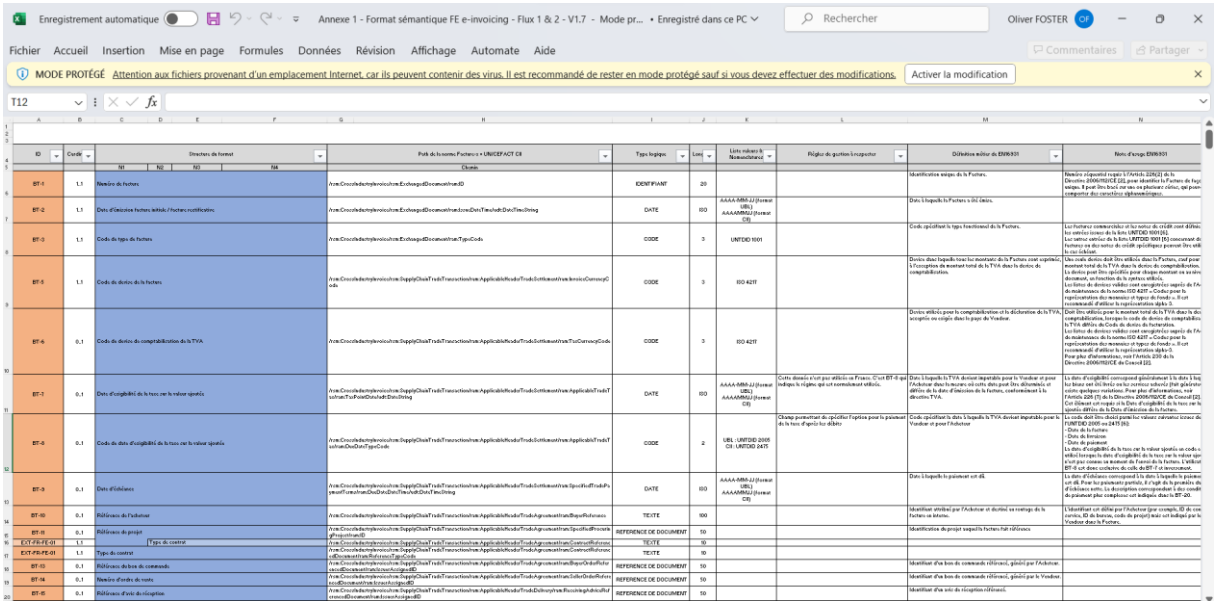
On peut trouver des exemples de fichiers sous format factur-x sur : https://fnfe-mpe.org/factur-x/factur-x-et-zugferd-2-2/

Documents indiquant la signification des balises :

https://www.impots.gouv.fr/specifications-externes-b2b

Version 2.3 du 31/07/2023 :

Télécharger les documents : Dossier de spécifications externes de la facturation électronique, annexes et swaggers (.zip)

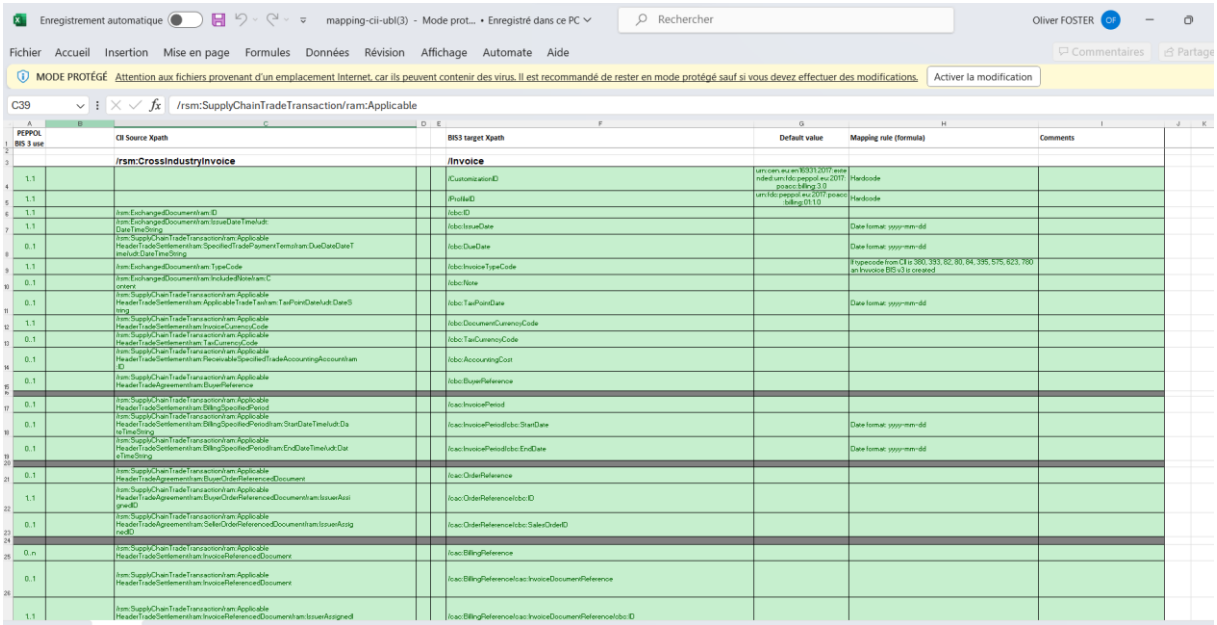


PEPPOL.eu
https://docs.peppol.eu › files › mapping-cii-ubl XLS

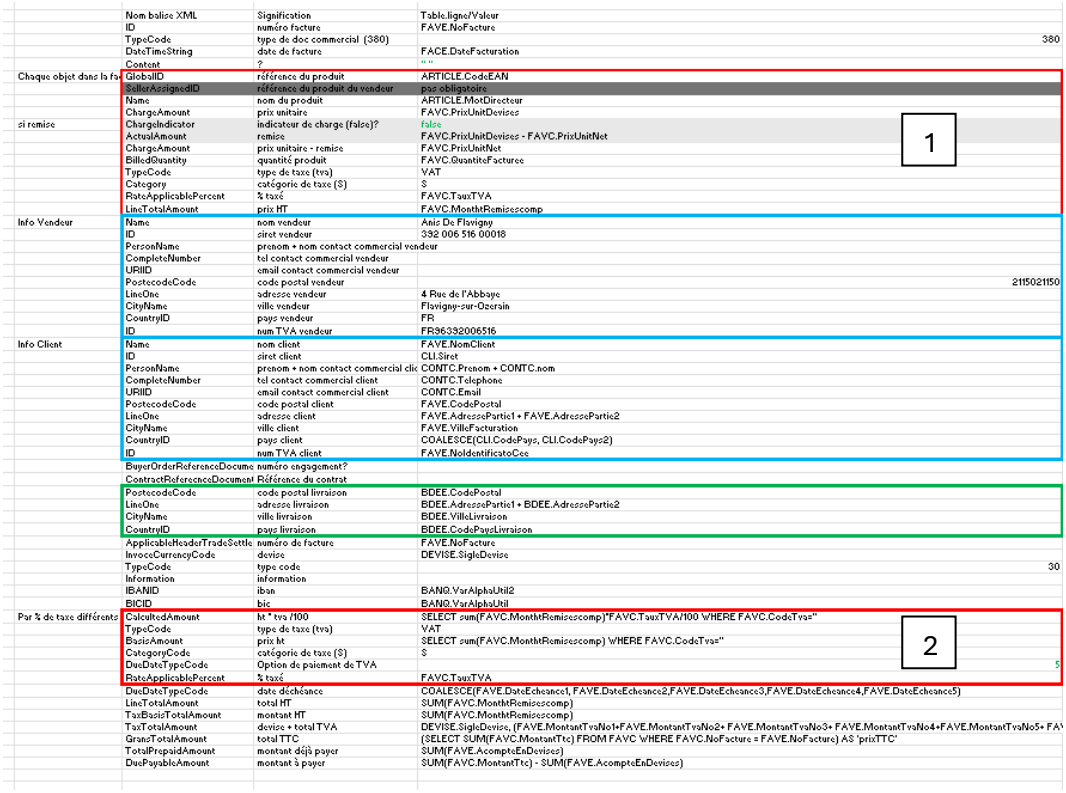
rsm:CrossIndustryInvoice /Invoice

1 janv. 1997 — **Mapping** rule (formula), Comments. 2. 3, /rsm:CrossIndustryInvoice, /Invoice ...
If not found in **CII** this is mapped from the line tax. 284, 0..1 ...

https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwigsrXMwqqEAxXsRaQEHd-_DO8QFnoECB0QAQ&url=https%3A%2F%2Fdocs.peppol.eu%2Fpoacc%2Fbilling%2F3.0%2Ffiles%2Fmapping-cii-ubl.xls&usg=AOvVaw1goOHboba12_6avsSXJg4w&opi=89978449



Recherches des valeurs dans les tables :



	Nom balise XML	Signification	Table.ligne/Valeur	
	ID	numéro facture	FAVE.NoFacture	
	TypeCode	type de doc commercial (380)		380
	DateTimeString	date de facture	FACE.DateFacturation	
	Content	?	""	
Chaque objet dans la fa	GlobalID	référence du produit	ARTICLE.CodeEAN	
	SellerAssignedID	référence du produit du vendeur	pas obligatoire	
	Name	nom du produit	ARTICLE.MotDirecteur	
	ChargeAmount	prix unitaire	FAVC.PrixUnitDevises	
si remise	ChargeIndicator	indicateur de charge (false)?	false	
	ActualAmount	remise	FAVC.PrixUnitDevises - FAVC.PrixUnitNet	1
	ChargeAmount	prix unitaire - remise	FAVC.PrixUnitNet	
	BilledQuantity	quantité produit	FAVC.QuantiteFacturee	
	TypeCode	type de taxe (tva)	VAT	
	Category	catégorie de taxe (S)	S	
	RateApplicablePercent	% taxé	FAVC.TauxTVA	
	LineTotalAmount	prix HT	FAVC.MonthtRemisescomp	
Info Vendeur	Name	nom vendeur	Anis De Flavigny	
	ID	siret vendeur	392 006 516 00018	
	PersonName	prenom + nom contact commercial vendeur		
	CompleteNumber	tel contact commercial vendeur		
	URIID	email contact commercial vendeur		
	PostcodeCode	code postal vendeur		2115021150
	LineOne	adresse vendeur	4 Rue de l'Abbaye	
	CityName	ville vendeur	Flavigny-sur-Ozerain	
	CountryID	pays vendeur	FR	
	ID	num TVA vendeur	FR36392006516	
Info Client	Name	nom client	FAVE.NomClient	
	ID	siret client	CLI.Siret	
	PersonName	prenom + nom contact commercial clie	CONTC.Prenom + CONTC.nom	
	CompleteNumber	tel contact commercial client	CONTC.Telephone	
	URIID	email contact commercial client	CONTC.Email	
	PostcodeCode	code postal client	FAVE.CodePostal	
	LineOne	adresse client	FAVE.AdressePartie1 + FAVE.AdressePartie2	
	CityName	ville client	FAVE.VilleFacturation	
	CountryID	pays client	COALESCE(CLI.CodePays, CLI.CodePays2)	
	ID	num TVA client	FAVE.NoIdentificatoCee	
	BuyerOrderReferenceDocume	numéro engagement?		
	ContractReferenceDocument	Référence du contrat		
	PostcodeCode	code postal livraison	BDEE.CodePostal	
	LineOne	adresse livraison	BDEE.AdressePartie1 + BDEE.AdressePartie2	
	CityName	ville livraison	BDEE.VilleLivraison	
	CountryID	pays livraison	BDEE.CodePaysLivraison	
	ApplicableHeaderTradeSettle	numéro de facture	FAVE.NoFacture	
	InvoceCurrencyCode	devise	DEVISE.SigleDevise	
	TypeCode	type code		30
	Information	information		
	IBANID	iban	BANQ.VarAlphaUtil2	
	BICID	bic	BANQ.VarAlphaUtil	
Par % de taxe différents	CalcultedAmount	ht * tva /100	SELECT sum(FAVC.MonthtRemisescomp)*FAVC.TauxTVA/100 WHERE FAVC.CodeTva=''	
	TypeCode	type de taxe (tva)	VAT	
	BasisAmount	prix ht	SELECT sum(FAVC.MonthtRemisescomp) WHERE FAVC.CodeTva=''	
	CategoryCode	catégorie de taxe (S)	S	2
	DueDateTypeCode	Option de paiement de TVA		5
	RateApplicablePercent	% taxé	FAVC.TauxTVA	
	DueDateTypeCode	date déchéance	COALESCE(FAVE.DateEchesance1, FAVE.DateEchesance2,FAVE.DateEchesance3,FAVE.DateEchesance4,FAVE.DateEchesance5)	
	LineTotalAmount	total HT	SUM(FAVC.MonthtRemisescomp)	
	TaxBasisTotalAmount	montant HT	SUM(FAVC.MonthtRemisescomp)	
	TaxTotalAmount	devise + total TVA	DEVISE.SigleDevise, (FAVE.MontantTvaNo1+FAVE.MontantTvaNo2+ FAVE.MontantTvaNo3+ FAVE.MontantTvaNo4+FAVE.MontantTvaNo5+ FA	
	GrandTotalAmount	total TTC	(SELECT SUM(FAVC.MontantTtc) FROM FAVC WHERE FAVC.NoFacture = FAVE.NoFacture) AS 'prixTTC'	
	TotalPrepaidAmount	montant déjà payer	SUM(FAVE.AcompteEnDevises)	
	DuePayableAmount	montant à payer	SUM(FAVC.MontantTtc) - SUM(FAVE.AcompteEnDevises)	

Les valeurs dans les cases rouges sont les valeurs qui se répètent.

1. Par produits
2. Par taux de tva (0%, 5,5%, 20%)

```xml
<ram:AssociatedDocumentLineDocument>
  <ram:LineID>1</ram:LineID>
</ram:AssociatedDocumentLineDocument>
<ram:SpecifiedTradeProduct>
  <ram:GlobalID schemeID="0160">3518370400049</ram:GlobalID>
  <ram:SellerAssignedID>NOUG250</ram:SellerAssignedID>
  <ram:Name>Nougat de l'Abbaye 250g</ram:Name>
</ram:SpecifiedTradeProduct>
<ram:SpecifiedLineTradeAgreement>
  <ram:GrossPriceProductTradePrice>
    <ram:ChargeAmount>4.55</ram:ChargeAmount>
    <ram:AppliedTradeAllowanceCharge>
      <ram:ChargeIndicator>
        <udt:Indicator>false</udt:Indicator>
      </ram:ChargeIndicator>
      <ram:ActualAmount>0.45</ram:ActualAmount>
    </ram:AppliedTradeAllowanceCharge>
  </ram:GrossPriceProductTradePrice>
  <ram:NetPriceProductTradePrice>
    <ram:ChargeAmount>4.10</ram:ChargeAmount>
  </ram:NetPriceProductTradePrice>
</ram:SpecifiedLineTradeAgreement>
<ram:SpecifiedLineTradeDelivery>
  <ram:BilledQuantity unitCode="C62">20.000</ram:BilledQuantity>
</ram:SpecifiedLineTradeDelivery>
<ram:SpecifiedLineTradeSettlement>
  <ram:ApplicableTradeTax>
    <ram:TypeCode>VAT</ram:TypeCode>
    <ram:CategoryCode>S</ram:CategoryCode>
    <ram:RateApplicablePercent>20.00</ram:RateApplicablePercent>
  </ram:ApplicableTradeTax>
  <ram:SpecifiedTradeSettlementLineMonetarySummation>
    <ram:LineTotalAmount>81.90</ram:LineTotalAmount>
  </ram:SpecifiedTradeSettlementLineMonetarySummation>
</ram:SpecifiedLineTradeSettlement>
</ram:IncludedSupplyChainTradeLineItem>
<ram:IncludedSupplyChainTradeLineItem>
<ram:AssociatedDocumentLineDocument>
  <ram:LineID>2</ram:LineID>
</ram:AssociatedDocumentLineDocument>
<ram:SpecifiedTradeProduct>
  <ram:GlobalID schemeID="0160">3518370200090</ram:GlobalID>
  <ram:SellerAssignedID>BRAIS300</ram:SellerAssignedID>
  <ram:Name>Biscuits aux raisins 300g</ram:Name>
</ram:SpecifiedTradeProduct>
<ram:SpecifiedLineTradeAgreement>
  <ram:GrossPriceProductTradePrice>
    <ram:ChargeAmount>3.20</ram:ChargeAmount>
  </ram:GrossPriceProductTradePrice>
  <ram:NetPriceProductTradePrice>
    <ram:ChargeAmount>3.20</ram:ChargeAmount>
  </ram:NetPriceProductTradePrice>
</ram:SpecifiedLineTradeAgreement>
<ram:SpecifiedLineTradeDelivery>
  <ram:BilledQuantity unitCode="C62">15.000</ram:BilledQuantity>
</ram:SpecifiedLineTradeDelivery>
<ram:SpecifiedLineTradeSettlement>
  <ram:ApplicableTradeTax>
    <ram:TypeCode>VAT</ram:TypeCode>
    <ram:CategoryCode>S</ram:CategoryCode>
    <ram:RateApplicablePercent>5.50</ram:RateApplicablePercent>
  </ram:ApplicableTradeTax>
  <ram:SpecifiedTradeSettlementLineMonetarySummation>
    <ram:LineTotalAmount>48.00</ram:LineTotalAmount>
  </ram:SpecifiedTradeSettlementLineMonetarySummation>
</ram:SpecifiedLineTradeSettlement>
</ram:IncludedSupplyChainTradeLineItem>
<ram:IncludedSupplyChainTradeLineItem>
<ram:AssociatedDocumentLineDocument>
  <ram:LineID>3</ram:LineID>
</ram:AssociatedDocumentLineDocument>
```

```xml
<ram:ApplicableTradeTax>
  <ram:CalculatedAmount>16.38</ram:CalculatedAmount>
  <ram:TypeCode>VAT</ram:TypeCode>
  <ram:BasisAmount>81.90</ram:BasisAmount>
  <ram:CategoryCode>S</ram:CategoryCode>
  <ram:DueDateTypeCode>5</ram:DueDateTypeCode>
  <ram:RateApplicablePercent>20.00</ram:RateApplicablePercent>
</ram:ApplicableTradeTax>
<ram:ApplicableTradeTax>
  <ram:CalculatedAmount>29.87</ram:CalculatedAmount>
  <ram:TypeCode>VAT</ram:TypeCode>
  <ram:BasisAmount>543.00</ram:BasisAmount>
  <ram:CategoryCode>S</ram:CategoryCode>
  <ram:DueDateTypeCode>5</ram:DueDateTypeCode>
  <ram:RateApplicablePercent>5.50</ram:RateApplicablePercent>
</ram:ApplicableTradeTax>
```

La commande SQL :

```
SELECT
    FAVE.NoFacture AS 'ID',
    CONVERT(VARCHAR, FAVE.DateFacturation, 112) AS 'DateFact',
    ARTICLE.CodeEAN AS 'GlobalID',
    ARTICLE.MotDirecteur AS 'NameItem',
    FAVC.PrixUnitDevises AS 'ChargeAmount',
    (FAVC.PrixUnitDevises - FAVC.PrixUnitNet) AS 'ReducedValue',
    FAVC.PrixUnitNet AS 'ChargeAmount2',
    FAVC.QuantiteFacturee AS 'BilledQuantity',
    FAVC.TauxTVA AS 'RateApplicablePercent',
    FAVC.MonthtRemisescomp AS 'LineTotalAmount',
    FAVE.NomClient AS 'NameClient',
    CLI.Siret AS 'SIRET',
    (CONTC.Prenom + ' ' + CONTC.nom) AS 'PersonName',
    CONTC.Telephone AS 'CompleteNumber',
    CONTC.Email AS 'URIID',
    FAVE.CodePostal AS 'PostecodeCode',
    (FAVE.AdressePartie1 + ' ' + FAVE.AdressePartie2) AS 'LineOne',
    FAVE.VilleFacturation AS 'CityName',
    COALESCE(CLI.CodePays, CLI.CodePays2) AS 'CountryID',
    CLI.NoIdentificatoCee AS 'numTVA',
    BDEE.CodePostal AS 'PostecodeCodeLivr',
    (BDEE.AdressePartie1 + ' ' + BDEE.AdressePartie2) AS 'LineOneLivr',
    BDEE.VilleLivraison AS 'CityNameLivr',
    BDEE.CodePaysLivraison AS 'CountryIDLivr',
    DEVISE.SigleDevise AS 'InvoiceCurrencyCode',
    BANQ.VarAlphaUtil2 AS 'IBANID',
    BANQ.VarAlphaUtil AS 'BICID',
    SUM(FAVC.MonthtRemisescomp) AS 'BasisAmount',
    CONVERT(VARCHAR, COALESCE(FAVE.DateEcheance1, FAVE.DateEcheance2, FAVE.DateEcheance3,
FAVE.DateEcheance4, FAVE.DateEcheance5), 112) AS 'DueDateTypeCode',
    (SELECT SUM(FAVC.PrixUnitDevises * FAVC.QuantiteFacturee) FROM FAVC WHERE
FAVC.NoFacture='FA24/00206' GROUP BY FAVC.NoFacture) AS 'LineTotalAmount2',
    FAVE.MontantTtc AS 'GrandTotalAmount',
    FAVE.AcompteEnDevises AS 'TotalPrepaidAmount',
    FAVE.MontantTtc-FAVE.AcompteEnDevises AS 'DuePayableAmount',
        CLI.CodeClient
FROM
    FAVE
JOIN
    FAVC ON FAVE.NoFacture = FAVC.NoFacture
JOIN
    ARTICLE ON ARTICLE.CodeArticle = FAVC.CodeArticleprest
JOIN
    CLI ON FAVE.CodeClient = CLI.CodeClient
JOIN
    CONTC ON CLI.VarAlphaUtil = CONTC.CodeUtilisateur
JOIN
    BDEE ON BDEE.NoBonExpedito = FAVC.NoBonExpedition
JOIN
    DEVISE ON FAVE.CodeDevise = DEVISE.CodeDevise
JOIN
    BANQ ON BANQ.CodeBanque = FAVE.CodeBanque1
WHERE
    FAVE.NoFacture = ?
    AND FAVC.PrestTalon = 'N'
GROUP BY
    FAVE.NoFacture, FAVE.DateFacturation, ARTICLE.CodeEAN, ARTICLE.MotDirecteur, FAVC.PrixUnitDevises,
FAVC.PrixUnitNet, FAVC.QuantiteFacturee, FAVC.TauxTVA, FAVE.DateEcheance1, FAVE.DateEcheance2,
FAVE.DateEcheance3, FAVE.DateEcheance4, FAVE.DateEcheance5, FAVC.MonthtRemisescomp, FAVE.NomClient,
CLI.Siret, CONTC.Prenom, CONTC.nom, CONTC.Telephone, CONTC.Email, FAVE.CodePostal, FAVE.AdressePartie1,
FAVE.AdressePartie2, FAVE.VilleFacturation, CLI.CodePays, CLI.CodePays2, CLI.NoIdentificatoCee,
BDEE.CodePostal, BDEE.AdressePartie1, BDEE.AdressePartie2, BDEE.VilleLivraison, BDEE.CodePaysLivraison,
DEVISE.SigleDevise, BANQ.VarAlphaUtil2, BANQ.VarAlphaUtil, FAVE.MontantTtc, FAVE.AcompteEnDevises,
CLI.CodeClient
ORDER BY
    FAVE.NoFacture DESC
```

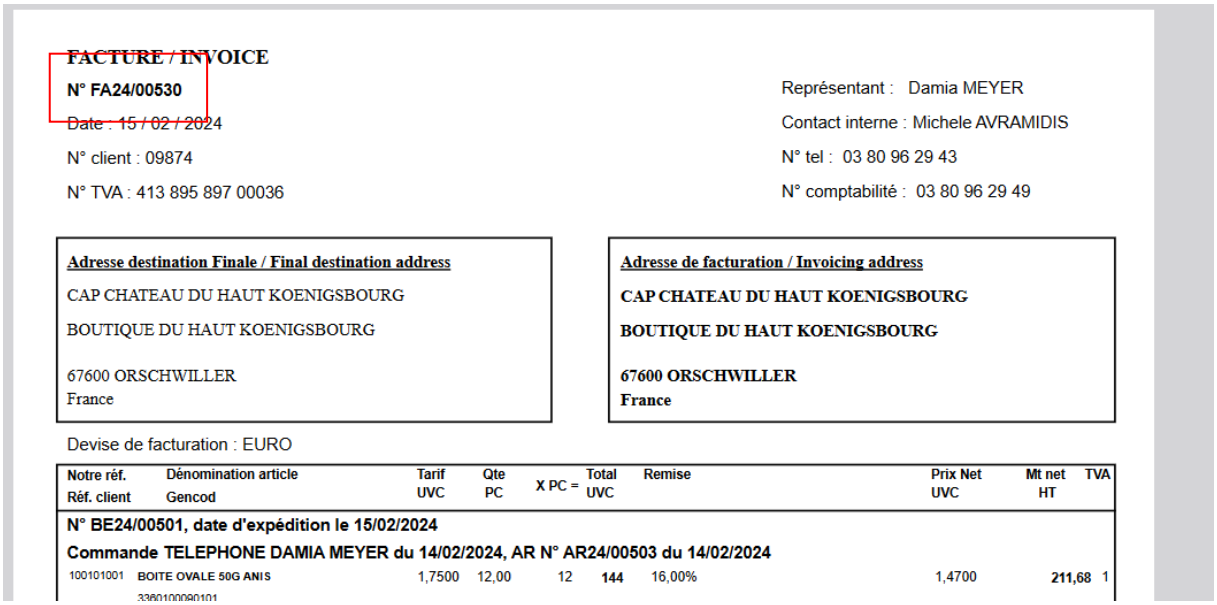
Voici un exemple :

```
SELECT
    FAVE.NoFacture AS 'ID',
    CONVERT(VARCHAR, FAVE.DateFacturation, 112) AS 'DateFact',
    ARTICLE.CodeEAN AS 'GlobalID',
    ARTICLE.MotDirecteur AS 'NameItem',
    FAVC.PrixUnitDevises AS 'ChargeAmount',
    (FAVC.PrixUnitDevises - FAVC.PrixUnitNet) AS 'ReducedValue',
    FAVC.PrixUnitNet AS 'ChargeAmount2',
    FAVC.QuantiteFacturee AS 'BilledQuantity',
    FAVC.TauxTVA AS 'RateApplicablePercent',
    FAVC.MontHtRemisescomp AS 'LineTotalAmount',
    FAVE.NomClient AS 'NameClient',
    CLI.Siret AS 'SIRET',
    (CONTC.Prenom + ' ' + CONTC.nom) AS 'PersonName',
    CONTC.Telephone AS 'CompleteNumber',
    CONTC.Email AS 'URID',
    FAVE.CodePostal AS 'PostecodeCode',
    (FAVE.AdressePartie1 + ' ' + FAVE.AdressePartie2) AS 'LineOne',
    FAVE.VilleFacturation AS 'CityName',
    COALESCE(CLI.CodePays, CLI.CodePays2) AS 'CountryID',
    CLI.NoIdentificatoCee AS 'numTVA',
    BDEE.CodePostal AS 'PostecodeCodeLivr',
    (BDEE.AdressePartie1 + ' ' + BDEE.AdressePartie2) AS 'LineOneLivr',
    BDEE.VilleLivraison AS 'CityNameLivr',
    BDEE.CodePaysLivraison AS 'CountryIDLivr',
    DEVISE.SigleDevise AS 'InvoiceCurrencyCode',
    BANQ.VarAlphaUtil2 AS 'IBANID',
    BANQ.VarAlphaUtil AS 'BICID',
    SUM(FAVC.MontHtRemisescomp) AS 'BasisAmount',
    CONVERT(VARCHAR, COALESCE(FAVE.DateEcheance1, FAVE.DateEcheance2, FAVE.DateEcheance3, FAVE.DateEcheance4, FAVE.DateEcheance5), 112) AS 'DueDateTypeCode',
    (SELECT SUM(FAVC.PrixUnitDevises * FAVC.QuantiteFacturee) FROM FAVC WHERE FAVC.NoFacture='FA24/00206' GROUP BY FAVC.NoFacture) AS 'LineTotalAmount',
    FAVE.MontantTtc AS 'GrandTotalAmount',
    FAVE.AcompteEnDevises AS 'TotalPrepaidAmount',
    FAVE.MontantTtc-FAVE.AcompteEnDevises AS 'DuePayableAmount'
```

Results | Messages

ID	DateFact	GlobalID	NameItem	ChargeAmount	ReducedValue	ChargeAmount2	BilledQuantity	RateApplicablePercent	LineTotalAmount	NameClient	SIRET	PersonName	CompleteNumber
FA24/00205	20240118	3360100090101	BOITE OVALE 50G ANIS	1.75000000	0.00000000	1.75000000	36.00000000	20.00000000	63.00000000	GAMM VERT SA - PARIS	33789128700019	Michele AVRAMIDIS	03 80 96 29 43
FA24/00205	20240118	3360100150102	BOITE OVALE 50G FLEUR ORANGER	1.75000000	0.00000000	1.75000000	24.00000000	20.00000000	42.00000000	GAMM VERT SA - PARIS	33789128700019	Michele AVRAMIDIS	03 80 96 29 43
FA24/00205	20240118	3360100180109	SAC 500G ANIS	9.15000000	0.00000000	9.15000000	4.00000000	20.00000000	36.60000000	GAMM VERT SA - PARIS	33789128700019	Michele AVRAMIDIS	03 80 96 29 43
FA24/00205	20240118	3360100520103	BOITE OVALE 50G CITRON	1.75000000	0.00000000	1.75000000	24.00000000	20.00000000	42.00000000	GAMM VERT SA - PARIS	33789128700019	Michele AVRAMIDIS	03 80 96 29 43
FA24/00205	20240118	3360102150100	BOITE OVALE 50G CASSIS	1.75000000	0.00000000	1.75000000	24.00000000	20.00000000	42.00000000	GAMM VERT SA - PARIS	33789128700019	Michele AVRAMIDIS	03 80 96 29 43
FA24/00205	20240118	3360102280012	BOITE OVALE 50G EUCALYPTUS BIOLOGIQUE	2.17000000	0.00000000	2.17000000	12.00000000	20.00000000	26.04000000	GAMM VERT SA - PARIS	33789128700019	Michele AVRAMIDIS	03 80 96 29 43

Comme on a maintenant les données, il faut récupérer le numéro de la facture. On parcourt le fichier pour retrouver du texte que ressemble à ce format : "FAXX/XXXXX"



FACTURE / INVOICE

N° FA24/00530

Date : 15 / 02 / 2024

N° client : 09874

N° TVA : 413 895 897 00036

Représentant : Damia MEYER

Contact interne : Michele AVRAMIDIS

N° tel : 03 80 96 29 43

N° comptabilité : 03 80 96 29 49

Adresse destination Finale / Final destination address

CAP CHATEAU DU HAUT KOENIGSBOURG

BOUTIQUE DU HAUT KOENIGSBOURG

67600 ORSCHWILLER
France

Adresse de facturation / Invoicing address

CAP CHATEAU DU HAUT KOENIGSBOURG

BOUTIQUE DU HAUT KOENIGSBOURG

67600 ORSCHWILLER
France

Devise de facturation : EURO

Notre réf. / Réf. client	Dénomination article / Gencod	Tarif UVC	Qte PC	X PC =	Total UVC	Remise	Prix Net UVC	Mt net HT	TVA
N° BE24/00501, date d'expédition le 15/02/2024									
Commande TELEPHONE DAMIA MEYER du 14/02/2024, AR N° AR24/00503 du 14/02/2024									
100101001	BOITE OVALE 50G ANIS 3360100090101	1,7500	12,00	12	144	16,00%	1,4700	211,68	1

```python
def getNoFacture(pdf_path,file_path,errorPath):
    text = ""
    with open(pdf_path, 'rb') as file:
        reader = PyPDF2.PdfReader(file)
        for i in range(len(reader.pages)):
            page = reader.pages[i]
            text += page.extract_text()
    if re.findall(r"FA \d{2}/\d{5}",text):
        return re.findall(r"FA \d{2}/\d{5}",text)[0].replace(' ','')
    else :
        destination_path = os.path.join(errorPath, os.path.basename(file_path))
        shutil.move(file_path, destination_path)
        print("err " + pdf_path + f"\n numéro de facture pas trouvé")
        logErr.error(pdf_path + f"\n numéro de facture pas trouvé")
        log.error(pdf_path + f"\n numéro de facture pas trouvé")
        return False
```

Puis on récupère les données sur python :

```
# Récupération des données à partir de la base de données
def getData(noFacture):
    # Commande SQL pour récupérer les données
    sqlCommand=''' commande sql
    '''

    server = 'PC-02095' #à modifier
    database = 'SILOG' #à modifier

    try:
        # Connexion à la base de données et exécution de la commande SQL
        conn_str = f'DRIVER={{SQL Server}};SERVER={server};DATABASE={database};Trusted_Connection=yes;'
        with pyodbc.connect(conn_str) as conn:
            cursor = conn.cursor()
            cursor.execute(sqlCommand,noFacture)
            rows = cursor.fetchall()
            # Formatage des données en une liste de dictionnaires
            result = []
            for row in rows:
                # Traitement de chaque ligne et ajout au résultat
                result.append({
                    # Mappage des colonnes de la base de données aux clés des dictionnaires
                    # et conversion des types si nécessaire
                    'ID': row[0], #numéro de la facture
                    'DateFact': row[1], #date de la facture
                    'GlobalID': row[2].replace(' ',''), #code ean du produit
                    'NameItem': checkString(row[3]), #nom du produit
                    'ChargeAmount': '%.2f' % float(row[4]), #prix produit
                    'ReducedValue': '%.2f' % float(row[5]), #remise
                    'ChargeAmountReduced': '%.2f' % float(row[6]), #prix produit - remise
                    'BilledQuantity': int(row[7]), #quantité de produit
                    'RateApplicablePercent': '%.2f' % float(row[8]), #% tva
                    'LineTotalAmount': '%.2f' % float(row[9]), # prix HT
                    'NameClient': checkString(row[10]), #nom du client
                    'SIRET': row[11].replace(' ',''), #siret du client
                    'PersonName': checkString(row[12]), #nom de la personne représentant le client
                    'CompleteNumber': row[13].replace(' ',''), # téléphone de la personne représentant
le client
                    'URIID': row[14], #email de la personne représentant le client
                    'PostecodeCode': row[15], #code postale du client
                    'LineOne': checkString(row[16]), #num porte + rue du client
                    'CityName': checkString(row[17]), #ville client
                    'CountryID': row[18], #num pays client
                    'numTVA': row[19].replace(' ',''), #numéro TVA
                    'PostecodeCodeLivr': checkString(row[20]), #code postale livraison
                    'LineOneLivr': checkString(row[21]), #num porte + rue de livraison
                    'CityNameLivr': checkString(row[22]), #nom ville de livraison
                    'CountryIDLivr': row[23], #num pays de livraison
                    'InvoiceCurrencyCode': row[24], #code devise
                    'IBANID': row[25].replace(' ',''), #iban
                    'BICID': row[26].replace(' ',''), #bic
                    'BasisAmount': '%.2f' % float(row[27]),# prix ht
                    'DueDateTypeCode': row[28], #data d'échéance
                    'TotalAmount': '%.2f' % float(row[29]), #prix tcc
                    'GrandTotalAmount': '%.2f' % float(row[30]), #prix ttc
                    'TotalPrepaidAmount': '%.2f' % float(row[31]), #quantité déjà payé
                    'DuePayableAmount': '%.2f' % float(row[32]), #quantité à payé
                    'CodeClient' : str(row[33])
                })
            return result
    except pyodbc.Error as ex:
        error_message = f"Erreur de connection à la base de données avec {noFacture}: {ex}"
        logErr.error(error_message)
        log.error(error_message)
        raise RuntimeError(error_message) from ex
    except Exception as ex:
        error_message = f"Une erreur est survenue avec {noFacture}: {ex}"
        logErr.error(error_message)
        log.error(error_message)
        raise RuntimeError(error_message) from ex
```

```
418     data = getData(noFacture)
419     print(data)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\temp\factures> & C:/Users/ofoster/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/temp/factures/getXML.py
[{'ID': 'FA24/00205', 'DateFact': '20240118', 'GlobalID': '3360100090101', 'NameItem': 'BOITE OVALE 50G ANIS', 'ChargeAmount': '1.75', 'ReducedValue': '0.00', 'ChargeAmountReduced': '1.75', 'BilledQuantity': 36, 'RateApplicablePercent': '20.00', 'LineTotalAmount': '63.00', 'NameClient': 'GAMM VERT SA - PARIS', 'SIRET': '33789128700019', 'PersonName': 'Michele AVRAMIDIS', 'CompleteNumber': '0380962943', 'URIID': 'commercial@anisdeflavigny.com', 'PostecodeCode': '77448', 'LineOne': 'TSA 14381 ', 'CityName': 'MARNE LA VALLEE CEDEX 02', 'CountryID': 'FR', 'numTVA': 'FR28832407704', 'PostecodeCodeLivr': '42110', 'LineOneLivr': '11 RUE DES PRAIRIES ZAC CARREFOUR', 'CityNameLivr': 'FEURS', 'CountryIDLivr': 'FR', 'InvoiceCurrencyCode': 'EUR', 'IBANID': 'FR7611006210100360960000130', 'BICID': 'AGRIFRPP810', 'BasisAmount': '63.00', 'DueDateTypeCode': '20240316', 'TotalAmount': '8329.44', 'GrandTotalAmount': '301.97', 'TotalPrepaidAmount': '0.00', 'DuePayableAmount': '301.97'}, {'ID': 'FA24/00205', 'DateFact': '20240118', 'GlobalID': '3360100150102', 'NameItem': 'BOITE OVALE 50G FLEUR ORANGER', 'ChargeAmount': '1.75', 'ReducedValue': '0.00', 'ChargeAmountReduced': '1.75', 'BilledQuantity': 24, 'RateApplicablePercent': '20.00', 'LineTotalAmount': '42.00', 'NameClient': 'GAMM VERT SA - PARIS', 'SIRET': '33789128700019', 'PersonName': 'Michele AVRAMIDIS', 'CompleteNumber': '0380962943', 'URIID': 'commercial@anisdeflavigny.com', 'PostecodeCode': '77448', 'LineOne': 'TSA 14381 ', 'CityName': 'MARNE LA VALLEE CEDEX 02', 'CountryID': 'FR', 'numTVA': 'FR28832407704', 'PostecodeCodeLivr': '42110', 'LineOneLivr': '11 RUE DES PRAIRIES ZAC CARREFOUR', 'CityNameLivr': 'FEURS', 'CountryIDLivr': 'FR', 'InvoiceCurrencyCode': 'EUR', 'IBANID': 'FR7611006210100360960000130', 'BICID': 'AGRIFRPP810', 'BasisAmount': '42.00', 'DueDateTypeCode': '20240316', 'TotalAmount': '8329.44', 'GrandTotalAmount': '301.97', 'TotalPrepaidAmount': '0.00', 'DuePayableAmount': '301.97'}, {'ID': 'FA24/00205', 'DateFact': '20240118', 'GlobalID': '3360100180109', 'NameItem': 'SAC 500G ANIS', 'ChargeAmount': '9.15', 'ReducedValue': '0.00', 'ChargeAmountReduced': '9.15', 'BilledQuantity': 4, 'RateApplicablePercent': '20.00', 'LineTotalAmount': '36.60', 'NameClient': 'GAMM VERT SA - PARIS', 'SIRET': '33789128700019', 'PersonName': 'Michele AVRAMIDIS', 'CompleteNumber': '0380962943', 'URIID': 'commercial@anisdeflavigny.com', 'PostecodeCode': '77448', 'LineOne': 'TSA 14381 ', 'CityName': 'MARNE LA VALLEE CEDEX 02', 'CountryID': 'FR', 'numTVA': 'FR28832407704', 'PostecodeCodeLivr': '42110', 'LineOneLivr': '11 RUE DES PRAIRIES ZAC CARREFOUR', 'CityNameLivr': 'FEURS', 'CountryIDLivr': 'FR', 'InvoiceCurrencyCode': 'EUR', 'IBANID': 'FR7611006210100360960000130', 'BICID': 'AGRIFRPP810', 'BasisAmount': '36.60', 'DueDateTypeCode': '20240316', 'TotalAmount': '8329.44', 'Grand
```

La variable `data` est un tableau de dictionnaires.

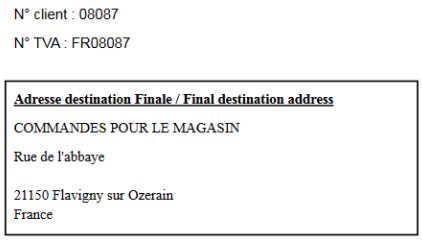
Chaque valeur dans la table représente un produit dans la facture.

Produit
ID
DateFact
GlobalID
NameItem
ChargeAmount
ReducedValue
ChargeAmountReduced
BilledQuantity
RateApplicablePercent
LineTotalAmount
NameClient
SIRET
PersonName
CompleteNumber
PostecodeCode
LineOne
Cityname
CountryID
numTVA
PostecodeCodeLivr
LineOneLivr
CityNameLivr
CountryIDLivr
InvoiceCurrencyCode
IBANID
BICID
BasisAmount
DueDateTypeCode
TotalAmount
GrandTotalAmount
TotalPrepaidAmount
DuePayableAmount
CodeClient

On fait passer les données par des fonctions pour vérifier la validité les données et de signaler l'utilisateur aux erreurs des valeurs dans la base.

```
# Vérification des données pour s'assurer qu'il n'y a pas de valeurs manquantes
def verifData(data, noFacture,file_path,errorPath):
    vide = []
    for i in range(len(data)):
        for key, val in data[i].items():
            if val == '':
                vide.append((i+1,key))
    if vide:
        error_message = f"{noFacture}: Il y a des données manquantes dans les lignes et colonnes:\n " +
'\n'.join([f"({idx}, '{key}')" for idx, key in vide])
        destination_path = os.path.join(errorPath, os.path.basename(file_path))
        shutil.move(file_path, destination_path)
        print("err " + error_message)
        logErr.error(error_message)
        log.error(error_message)
        return False
    return True

def verifFormat(data, noFacture, file_path, errorPath):
    emailFormat = re.compile(r'^[\w\.-]+@[\w\.-]+\.\w+$')
    destination_path = os.path.join(errorPath, os.path.basename(file_path))

    for entry in data:
        if (len(entry['GlobalID']) != 13) or not(entry['GlobalID'].isnumeric()):
            shutil.move(file_path, destination_path)
            error_message = f"{noFacture},{entry['GlobalID']}, le format de l'EAN est invalide, il doit être composé
de 13 chiffres"
            print("err " +error_message)
            logErr.error(error_message)
            log.error(error_message)
            return False
        if (len(entry['SIRET']) != 14) or not(entry['SIRET'].isnumeric()):
            shutil.move(file_path, destination_path)
            error_message = f"{noFacture},{entry['SIRET']}, le format du SIRET est invalide, il doit être composé de
14 chiffres"
            print("err " +error_message)
            logErr.error(error_message)
            log.error(error_message)
            return False
        if (len(entry['CompleteNumber']) != 10) or not(entry['CompleteNumber'].isnumeric()):
            shutil.move(file_path, destination_path)
            error_message = f"{noFacture},{entry['CompleteNumber']}, le format du numéro de téléphone du client est
invalide, il doit être composé de 10 chiffres"
            print("err " + error_message)
            logErr.error(error_message)
            log.error(error_message)
            return False
        if not(emailFormat.match(entry['URIID'])):
            shutil.move(file_path, destination_path)
            error_message = f"{noFacture},{entry['URIID']}, le format de l'adresse e-mail du client est invalide"
            print("err " +error_message)
            logErr.error(error_message)
            log.error(error_message)
            return False
        if not(entry['numTVA'][0:2].isalpha()) or len(entry['numTVA']) < 8:
            shutil.move(file_path, destination_path)
            error_message = f"{noFacture},{entry['numTVA']} le format du numéro de TVA est invalide"
            print("err " +error_message)
            logErr.error(error_message)
            log.error(error_message)
            return False
        if not(entry['IBANID'][0:2].isalpha()) or len(entry['IBANID']) < 14 or len(entry['IBANID']) > 34:
            shutil.move(file_path, destination_path)
            error_message = f"{noFacture},{entry['IBANID']} le format de l'IBAN est invalide"
            print("err " +error_message)
            logErr.error(error_message)
            log.error(error_message)
            return False
        if len(entry['BICID']) < 8 or len(entry['BICID']) > 11:
            shutil.move(file_path, destination_path)
            error_message = f"{noFacture},{entry['BICID']} le format du numéro BIC est invalide"
            print("err " +error_message)
            logErr.error(error_message)
            log.error(error_message)
            return False
    return True
```

L’entreprise envoie certaines factures à lui-même, donc on ne va pas générer le fichier xml, on va le déplacer le fichier PDF dans un dossier


N° client : 08087

N° TVA : FR08087

Adresse destination Finale / Final destination address

COMMANDES POUR LE MAGASIN

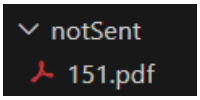
Rue de l'abbaye

21150 Flavigny sur Ozerain
France

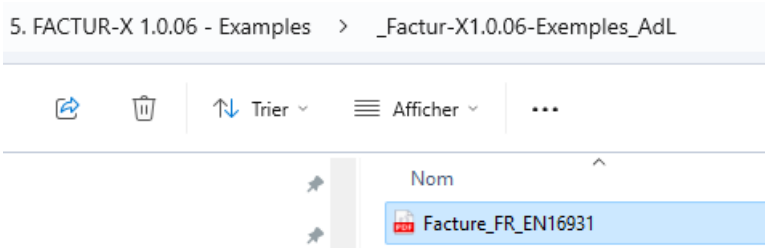
Donc on crée aussi cette fonction suivante :

```python
def verifClient(data,file_path):
    notSentPath = r"C:\temp\v4\factures\notSent"
    data = data[0]
    if data['CodeClient'] == '08087':
        shutil.move(file_path, notSentPath)
        return False
    else:
        return True
```

Exemple :



Ensuite, nous allons insérer les données dans des balises XML en utilisant ce fichier comme format :



On va tout d’abord gérer les balises qui se répète.

Pour les produits, on parcourt la liste `data` et on place les données dans les balises xml:

```
def setProduit(data):
    xmlProd = ''
    for i, row in enumerate(data):
        xmlProd += f'''
            <ram:IncludedSupplyChainTradeLineItem>
                <ram:AssociatedDocumentLineDocument>
                    <ram:LineID>{i+1}</ram:LineID>
                </ram:AssociatedDocumentLineDocument>
                <ram:SpecifiedTradeProduct>
                    <ram:GlobalID schemeID="0160">{row['GlobalID']}</ram:GlobalID>
                    <ram:Name>{row['NameItem']}</ram:Name>
                </ram:SpecifiedTradeProduct>
                <ram:SpecifiedLineTradeAgreement>
                    <ram:GrossPriceProductTradePrice>
                        <ram:ChargeAmount>{row['ChargeAmount']}</ram:ChargeAmount>
            '''
        if row['ReducedValue'] != '0.00':
            xmlProd += f'''
                <ram:AppliedTradeAllowanceCharge>
                    <ram:ChargeIndicator>
                        <udt:Indicator>false</udt:Indicator>
                    </ram:ChargeIndicator>
                    <ram:ActualAmount>{row['ReducedValue']}</ram:ActualAmount>
                </ram:AppliedTradeAllowanceCharge>
            '''
        xmlProd +=f'''
                    </ram:GrossPriceProductTradePrice>
                    <ram:NetPriceProductTradePrice>
                        <ram:ChargeAmount>{row['ChargeAmountReduced']}</ram:ChargeAmount>
                    </ram:NetPriceProductTradePrice>
                </ram:SpecifiedLineTradeAgreement>
                <ram:SpecifiedLineTradeDelivery>
                    <ram:BilledQuantity
unitCode="C62">{int(row['BilledQuantity'])}</ram:BilledQuantity>
                </ram:SpecifiedLineTradeDelivery>
                <ram:SpecifiedLineTradeSettlement>
                    <ram:ApplicableTradeTax>
                    <ram:TypeCode>VAT</ram:TypeCode>
                    <ram:CategoryCode>S</ram:CategoryCode>
                    <ram:RateApplicablePercent>{'%.2f' %
float(row['RateApplicablePercent'])}</ram:RateApplicablePercent>
                    </ram:ApplicableTradeTax>
                    <ram:SpecifiedTradeSettlementLineMonetarySummation>
                    <ram:LineTotalAmount>{row['LineTotalAmount']}</ram:LineTotalAmount>
                    </ram:SpecifiedTradeSettlementLineMonetarySummation>
                </ram:SpecifiedLineTradeSettlement>
            </ram:IncludedSupplyChainTradeLineItem>
        '''
    return xmlProd
```

Pour les taux de tva, c'est plus complexe, on répète chaque balise par le taux de tva différent.

Ex :

```
<ram:ApplicableTradeTax>
  <ram:CalculatedAmount>16.38</ram:CalculatedAmount>
  <ram:TypeCode>VAT</ram:TypeCode>
  <ram:BasisAmount>81.90</ram:BasisAmount>
  <ram:CategoryCode>S</ram:CategoryCode>
  <ram:DueDateTypeCode>5</ram:DueDateTypeCode>
  <ram:RateApplicablePercent>20.00</ram:RateApplicablePercent>
</ram:ApplicableTradeTax>
<ram:ApplicableTradeTax>
  <ram:CalculatedAmount>29.87</ram:CalculatedAmount>
  <ram:TypeCode>VAT</ram:TypeCode>
  <ram:BasisAmount>543.00</ram:BasisAmount>
  <ram:CategoryCode>S</ram:CategoryCode>
  <ram:DueDateTypeCode>5</ram:DueDateTypeCode>
  <ram:RateApplicablePercent>5.50</ram:RateApplicablePercent>
</ram:ApplicableTradeTax>
```

BasisAmount : somme du prix HT par taux de tva

CalculatedAmount : (BasisAmount*taux_tva)/100, montant TVA

```python
def getTVA(data):
    vat_dict = {}  # Dictionnaire pour stocker les totaux TVA par type
    total_taxed_amount = 0  # Initialisation du montant total taxé à zéro
    # Parcours de chaque ligne de données
    for row in data:
        # Obtention du type de TVA en pourcentage avec 4 décimales
        vat_type = '%.4f' % float(row['RateApplicablePercent'])
        # Calcul du montant de la taxe pour le produit actuel
        prixHT = float(row['LineTotalAmount'])
        tax_amount = prixHT * float(vat_type) / 100
        # Ajout du montant de la taxe au montant total taxé
        total_taxed_amount += tax_amount
        # Si le type de TVA existe déjà dans le dictionnaire, mettez à jour ses valeurs
        if vat_type in vat_dict:
            vat_dict[vat_type]['SumProduits'] += prixHT
            vat_dict[vat_type]['TotalTVA'] += tax_amount
        else:
            # Si le type de TVA n'existe pas dans le dictionnaire, ajoutez-le avec des
valeurs initiales
            vat_dict[vat_type] = {
                'SumProduits': prixHT,
                'TotalTVA': tax_amount
            }
    # Retourne le dictionnaire de TVA et le montant total taxé
    return vat_dict, total_taxed_amount
```

```
PS C:\temp\factures> & C:/Users/ofoster/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/temp/factures/getXML.py
({'20.0000': {'SumProduits': 251.64, 'TotalTVA': 50.327999999999996}}, 50.327999999999996)
PS C:\temp\factures>
```

Type TVA :
SumProduits
TotalTVA
Total TVA

Fonction pour créer les balises :

```
def setTva(data):
    xmlTVA = ''
    # Obtenir les types de TVA et les valeurs associées pour les données fournies
    typeTVA = getTVA(data)[0]
    # Parcours de chaque type de TVA et de ses valeurs
    for vat_type, values in typeTVA.items():
        # Construction de la section XML pour chaque type de TVA
        xmlTVA += f'''
            <ram:ApplicableTradeTax>
                <ram:CalculatedAmount>{'%.2f' %
float(values['TotalTVA'])}</ram:CalculatedAmount>
                <ram:TypeCode>VAT</ram:TypeCode>
                <ram:BasisAmount>{values['SumProduits']}</ram:BasisAmount>
                <ram:CategoryCode>S</ram:CategoryCode>
                <ram:DueDateTypeCode>5</ram:DueDateTypeCode>
                <ram:RateApplicablePercent>{vat_type}</ram:RateApplicablePercent>
            </ram:ApplicableTradeTax>
        '''
    return xmlTVA
```

On a donc tous ce dont on a besoin pour générer le fichier xml.

```
def setXML(data):
    #récupérer la valeur de TVA
    valTVA = getTVA(data)[1]
    row = data[0]
    xmlCode=f'''<?xml version='1.0' encoding='UTF-8'?>
    <rsm:CrossIndustryInvoice
xmlns:qdt="urn:un:unece:uncefact:data:standard:QualifiedDataType:100"
xmlns:ram="urn:un:unece:uncefact:data:standard:ReusableAggregateBusinessInformationEntity:1
00" xmlns:rsm="urn:un:unece:uncefact:data:standard:CrossIndustryInvoice:100"
xmlns:udt="urn:un:unece:uncefact:data:standard:UnqualifiedDataType:100"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <rsm:ExchangedDocumentContext>
            <ram:GuidelineSpecifiedDocumentContextParameter>
            <ram:ID>urn:cen.eu:en16931:2017</ram:ID>
            </ram:GuidelineSpecifiedDocumentContextParameter>
        </rsm:ExchangedDocumentContext>
        <rsm:ExchangedDocument>
            <ram:ID>{row['ID']}</ram:ID>
            <ram:TypeCode>380</ram:TypeCode>
            <ram:IssueDateTime>
            <udt:DateTimeString format="102">{row['DateFact']}</udt:DateTimeString>
            </ram:IssueDateTime>
            <ram:IncludedNote>
            <ram:Content></ram:Content>
            </ram:IncludedNote>
        </rsm:ExchangedDocument>
        <rsm:SupplyChainTradeTransaction>
'''
    xmlCode += setProduit(data)

    xmlCode += f'''
<ram:ApplicableHeaderTradeAgreement>
    <ram:SellerTradeParty>
        <ram:Name>Anis De Flavigny</ram:Name>
        <ram:SpecifiedLegalOrganization>
        <ram:ID schemeID="0002">45739641818284</ram:ID>
    '''
    #remplacer siret par celui du vendeur
```

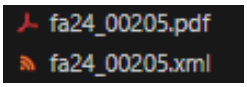
```
    xmlCode += f'''
        </ram:SpecifiedLegalOrganization>
        <ram:DefinedTradeContact>
        <ram:PersonName></ram:PersonName>
        <ram:TelephoneUniversalCommunication>
            <ram:CompleteNumber></ram:CompleteNumber>
        </ram:TelephoneUniversalCommunication>
        <ram:EmailURIUniversalCommunication>
            <ram:URIID schemeID="SMTP"></ram:URIID>
        </ram:EmailURIUniversalCommunication>
        </ram:DefinedTradeContact>
        <ram:PostalTradeAddress>
        <ram:PostcodeCode>21150</ram:PostcodeCode>
        <ram:LineOne>4 Rue de l'Abbaye</ram:LineOne>
        <ram:CityName>Flavigny-sur-Ozerain</ram:CityName>
        <ram:CountryID>FR</ram:CountryID>
        </ram:PostalTradeAddress>
        <ram:SpecifiedTaxRegistration>
        <ram:ID schemeID="VA">FR96392006516</ram:ID>
        </ram:SpecifiedTaxRegistration>
    </ram:SellerTradeParty>
    <ram:BuyerTradeParty>
        <ram:Name>{row['NameClient']}</ram:Name>
        <ram:SpecifiedLegalOrganization>
        <ram:ID schemeID="0002">12790703541348</ram:ID>
        '''
    #remplacer siret par celui du client
    xmlCode += f'''
        </ram:SpecifiedLegalOrganization>
        <ram:DefinedTradeContact>
        <ram:PersonName>{row['PersonName']}</ram:PersonName>
        <ram:TelephoneUniversalCommunication>
            <ram:CompleteNumber>{row['CompleteNumber']}</ram:CompleteNumber>
        </ram:TelephoneUniversalCommunication>
        <ram:EmailURIUniversalCommunication>
            <ram:URIID schemeID="SMTP">{row['URIID']}</ram:URIID>
        </ram:EmailURIUniversalCommunication>
        </ram:DefinedTradeContact>
        <ram:PostalTradeAddress>
            <ram:PostcodeCode>{row['PostecodeCode']}</ram:PostcodeCode>
            <ram:LineOne>{row['LineOne']}</ram:LineOne>
            <ram:CityName>{row['CityName']}</ram:CityName>
            <ram:CountryID>{row['CountryID']}</ram:CountryID>
        </ram:PostalTradeAddress>
        <ram:SpecifiedTaxRegistration>
        <ram:ID schemeID="VA">{row['numTVA']}</ram:ID>
        </ram:SpecifiedTaxRegistration>
    </ram:BuyerTradeParty>
    <ram:BuyerOrderReferencedDocument>
        <ram:IssuerAssignedID></ram:IssuerAssignedID>
    </ram:BuyerOrderReferencedDocument>
    <ram:ContractReferencedDocument>
        <ram:IssuerAssignedID></ram:IssuerAssignedID>
    </ram:ContractReferencedDocument>
</ram:ApplicableHeaderTradeAgreement>
<ram:ApplicableHeaderTradeDelivery>
    <ram:ShipToTradeParty>
        <ram:PostalTradeAddress>
        <ram:PostcodeCode>{row['PostecodeCodeLivr']}</ram:PostcodeCode>
        <ram:LineOne>{row['LineOneLivr']}</ram:LineOne>
        <ram:CityName>{row['CityNameLivr']}</ram:CityName>
        <ram:CountryID>{row['CountryIDLivr']}</ram:CountryID>
        </ram:PostalTradeAddress>
    </ram:ShipToTradeParty>
</ram:ApplicableHeaderTradeDelivery>
<ram:ApplicableHeaderTradeSettlement>
```

```
    <ram:PaymentReference>{row['ID']}</ram:PaymentReference>
    <ram:InvoiceCurrencyCode>{row['InvoiceCurrencyCode']}</ram:InvoiceCurrencyCode>
    <ram:SpecifiedTradeSettlementPaymentMeans>
        <ram:TypeCode>30</ram:TypeCode>
        <ram:Information></ram:Information>
        <ram:PayeePartyCreditorFinancialAccount>
            <ram:IBANID>{row['IBANID']}</ram:IBANID>
        </ram:PayeePartyCreditorFinancialAccount>
        <ram:PayeeSpecifiedCreditorFinancialInstitution>
        <ram:BICID>{row['BICID']}</ram:BICID>
        </ram:PayeeSpecifiedCreditorFinancialInstitution>
    </ram:SpecifiedTradeSettlementPaymentMeans>
    '''
    xmlCode += setTva(data)

    xmlCode +=f'''
            <ram:SpecifiedTradePaymentTerms>
                <ram:Description></ram:Description>
                <ram:DueDateDateTime>
                    <udt:DateTimeString
format="102">{row['DueDateTypeCode']}</udt:DateTimeString>
                </ram:DueDateDateTime>
            </ram:SpecifiedTradePaymentTerms>
            <ram:SpecifiedTradeSettlementHeaderMonetarySummation>
                <ram:LineTotalAmount>{'%.2f' % (float(row['GrandTotalAmount']) -
valTVA)}</ram:LineTotalAmount>
                <ram:TaxBasisTotalAmount>{'%.2f' % (float(row['GrandTotalAmount']) -
valTVA)}</ram:TaxBasisTotalAmount>
                <ram:TaxTotalAmount currencyID="{row['InvoiceCurrencyCode']}">{'%.2f' %
valTVA}</ram:TaxTotalAmount>
                <ram:GrandTotalAmount>{'%.2f' %
float(row['GrandTotalAmount'])}</ram:GrandTotalAmount>
                <ram:TotalPrepaidAmount>{'%.2f' %
float(row['TotalPrepaidAmount'])}</ram:TotalPrepaidAmount>
                <ram:DuePayableAmount>{'%.2f' %
float(row['DuePayableAmount'])}</ram:DuePayableAmount>
            </ram:SpecifiedTradeSettlementHeaderMonetarySummation>
        </ram:ApplicableHeaderTradeSettlement>
    </rsm:SupplyChainTradeTransaction>
</rsm:CrossIndustryInvoice>
    '''
    return xmlCode
```

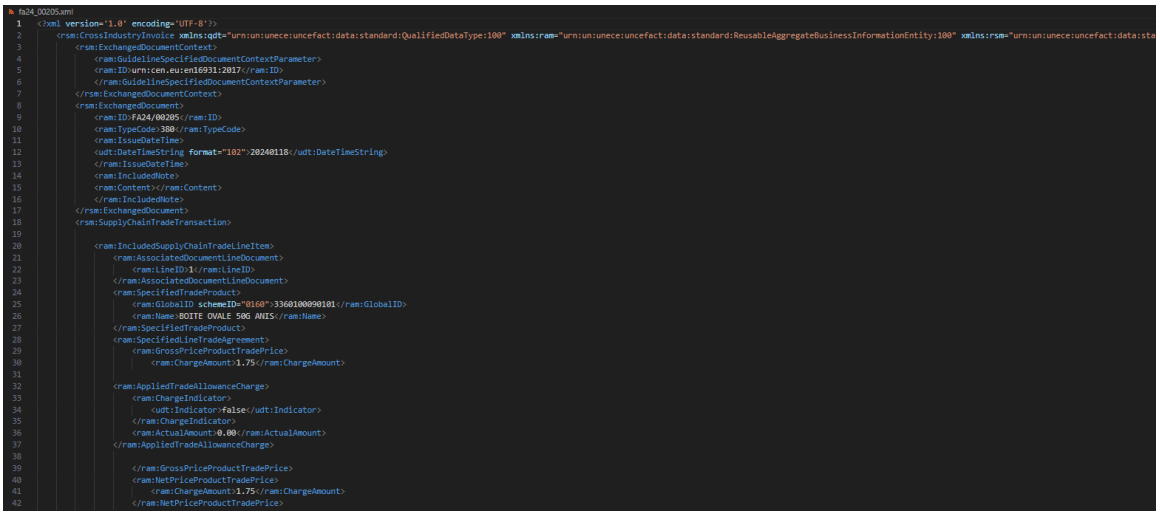
Finalement, Il faut générer le fichier xml

```
xmlCode = setXML(data)
with open(xmlPath, 'w', encoding='utf-8') as xmlFile:
      xmlFile.write(xmlCode)
```

Résultat :



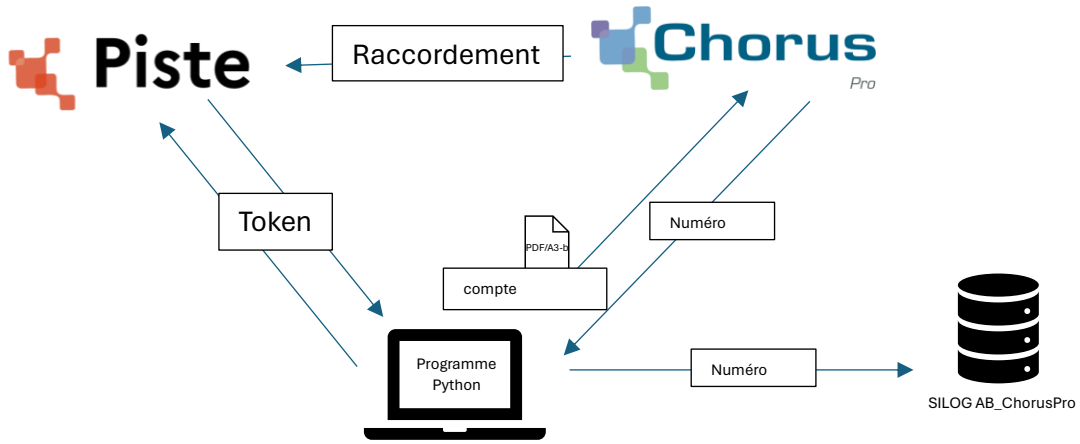
fa24_00205.xml



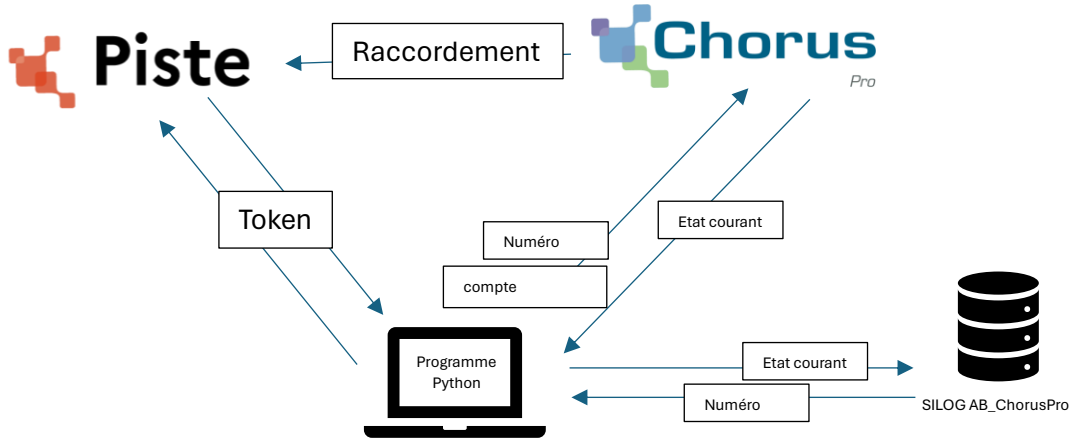
```xml
<?xml version='1.0' encoding='UTF-8'?>
<rsm:CrossIndustryInvoice xmlns:qdt="urn:un:unece:uncefact:data:standard:QualifiedDataType:100" xmlns:ram="urn:un:unece:uncefact:data:standard:ReusableAggregateBusinessInformationEntity:100" xmlns:rsm="urn:un:unece:uncefact:data:sta
    <rsm:ExchangedDocumentContext>
        <ram:GuidelineSpecifiedDocumentContextParameter>
        <ram:ID>urn:cen.eu:en16931:2017</ram:ID>
        </ram:GuidelineSpecifiedDocumentContextParameter>
    </rsm:ExchangedDocumentContext>
    <rsm:ExchangedDocument>
        <ram:ID>FA24/00205</ram:ID>
        <ram:TypeCode>380</ram:TypeCode>
        <ram:IssueDateTime>
        <udt:DateTimeString format="102">20240118</udt:DateTimeString>
        </ram:IssueDateTime>
        <ram:IncludedNote>
        <ram:Content></ram:Content>
        </ram:IncludedNote>
    </rsm:ExchangedDocument>
    <rsm:SupplyChainTradeTransaction>

        <ram:IncludedSupplyChainTradeLineItem>
            <ram:AssociatedDocumentLineDocument>
                <ram:LineID>1</ram:LineID>
            </ram:AssociatedDocumentLineDocument>
            <ram:SpecifiedTradeProduct>
                <ram:GlobalID schemeID="0160">3360100090101</ram:GlobalID>
                <ram:Name>BOITE OVALE 50G ANIS</ram:Name>
            </ram:SpecifiedTradeProduct>
            <ram:SpecifiedLineTradeAgreement>
                <ram:GrossPriceProductTradePrice>
                    <ram:ChargeAmount>1.75</ram:ChargeAmount>

            <ram:AppliedTradeAllowanceCharge>
                <ram:ChargeIndicator>
                    <udt:Indicator>false</udt:Indicator>
                </ram:ChargeIndicator>
                <ram:ActualAmount>0.00</ram:ActualAmount>
            </ram:AppliedTradeAllowanceCharge>

                </ram:GrossPriceProductTradePrice>
                <ram:NetPriceProductTradePrice>
                    <ram:ChargeAmount>1.75</ram:ChargeAmount>
                </ram:NetPriceProductTradePrice>
```

3) Mise en place d'un API

sendFact.py :



getEtatCoutant.py :



Prérequis :

```
pip install requests
pip install logging
pip install dateTime
pip install PyPDF2
pip install pyodbc
```

Microsoft SQL server management

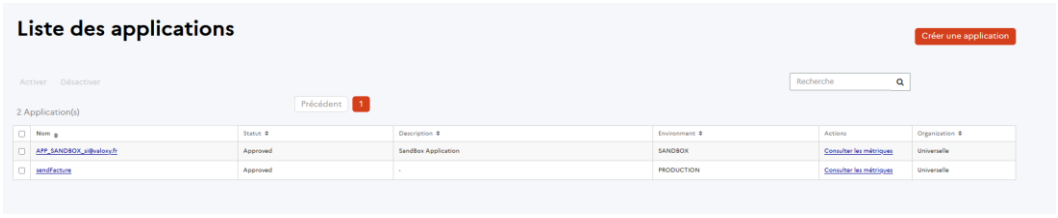
1) Création de l'api sur PISTE

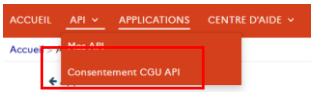
https://developer.aife.economie.gouv.fr/

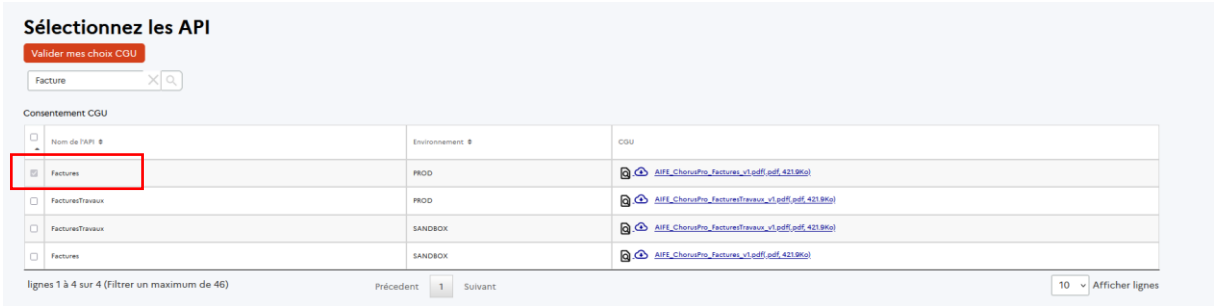
- Créer un compte/se connecter
- Dans la page « applications » :

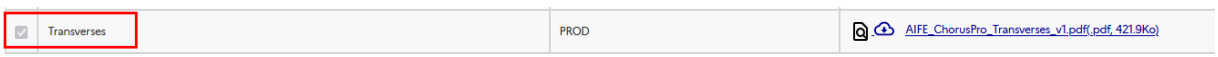
Puis Il faut remplir ce formulaire :

Dans ce cas on appelle l'API « sendFacture »

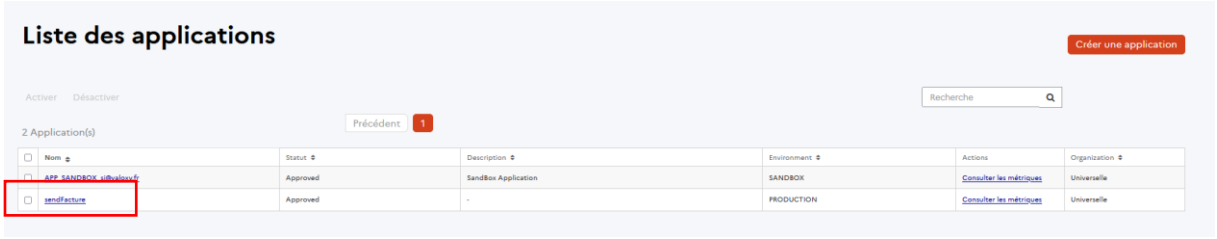




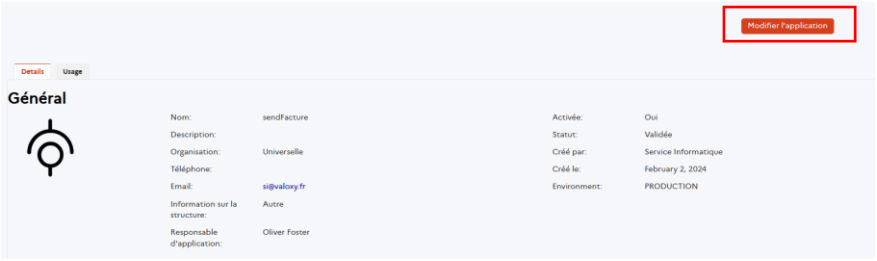




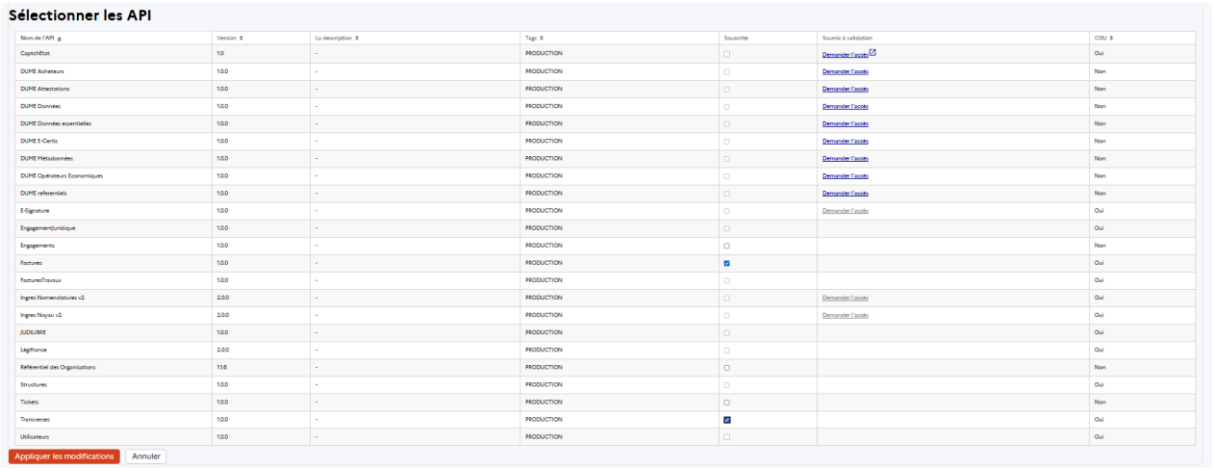
On revient dans la page « Applications » :



Consultation de l'application :
sendFacture



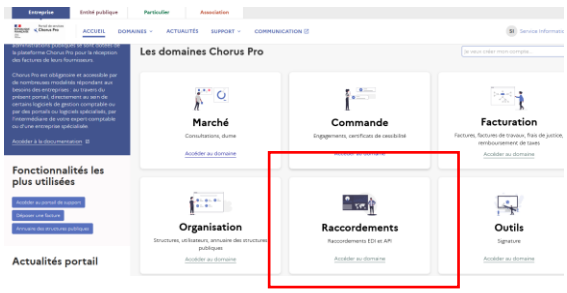
En bas de la page, cocher « factures » et « transverses » :

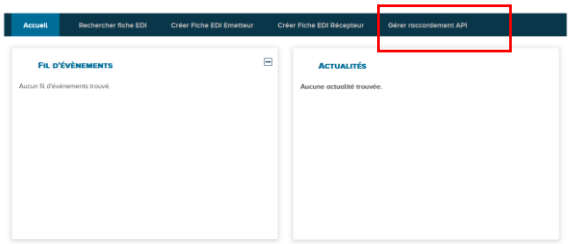


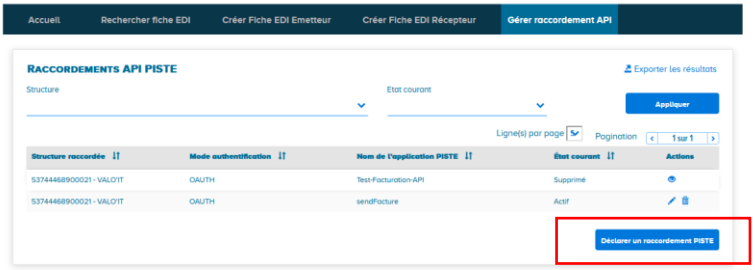
2) Raccordement sur Chorus Pro

a. Raccordé l'api PISTE sur chorus pro

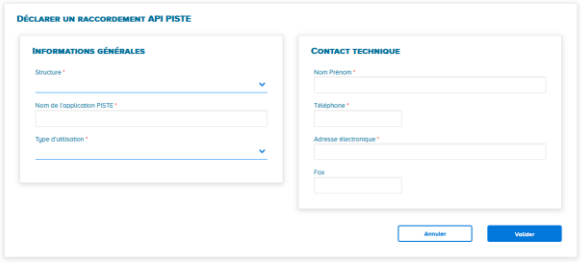
- Se connecter/créer un compte





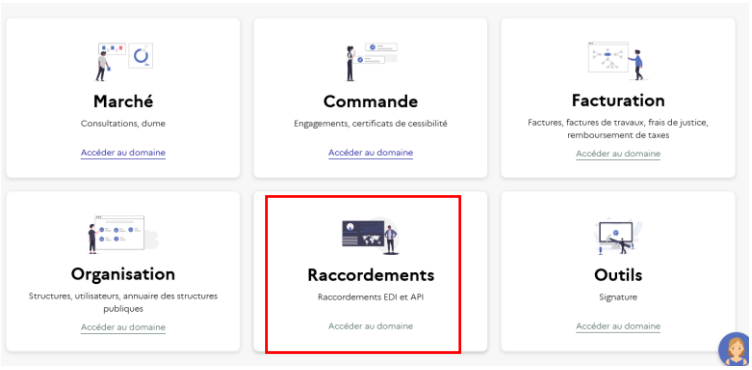


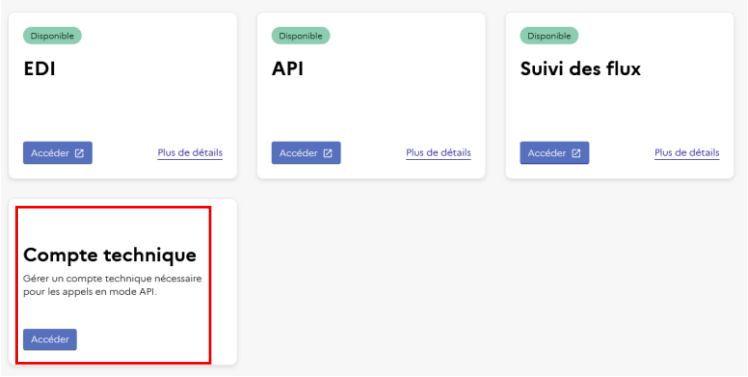
Remplir ce formulaire :



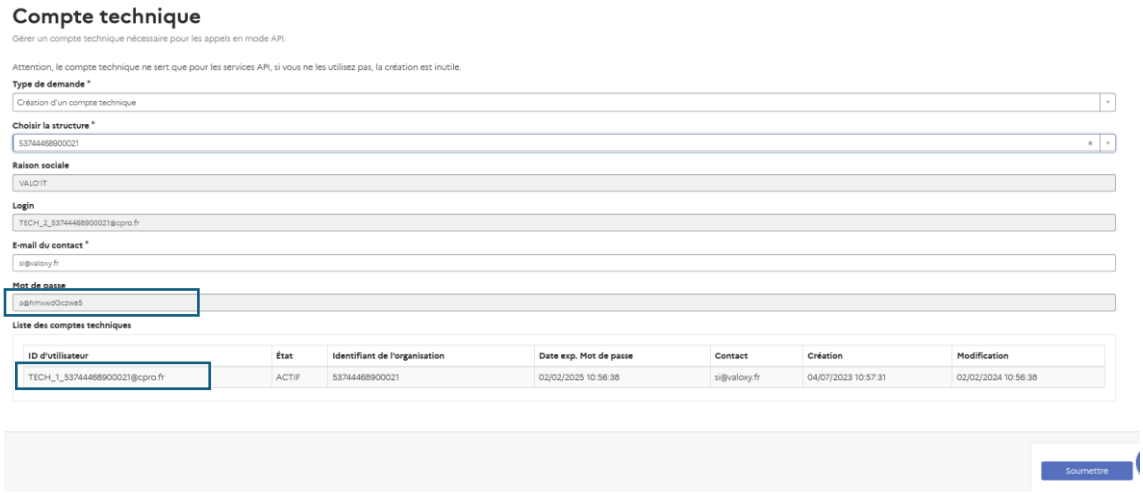
On a cela :

b) Création d’un compte technique





Remplir ce formulaire et récupérer les données ci-dessous

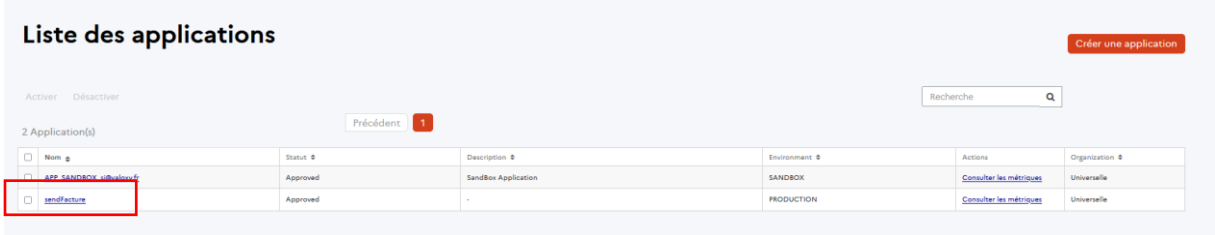


3) Créer l'API

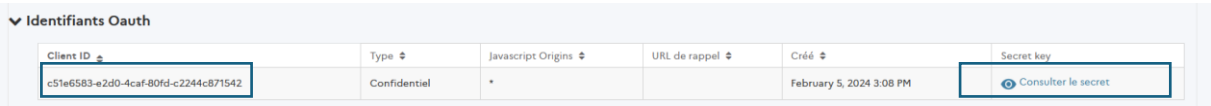
a. Récupérer le token (Authentification Oauth)

Sur PISTE :

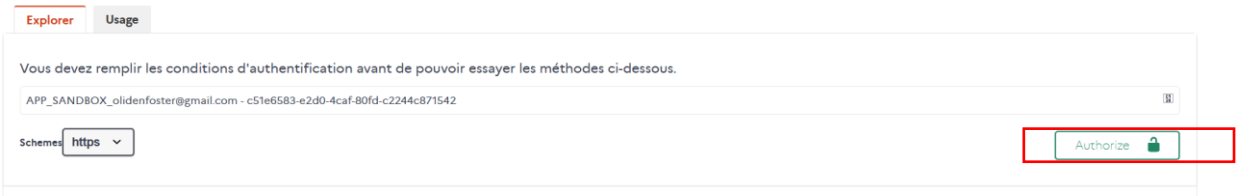
On revient dans la page « Applications » :



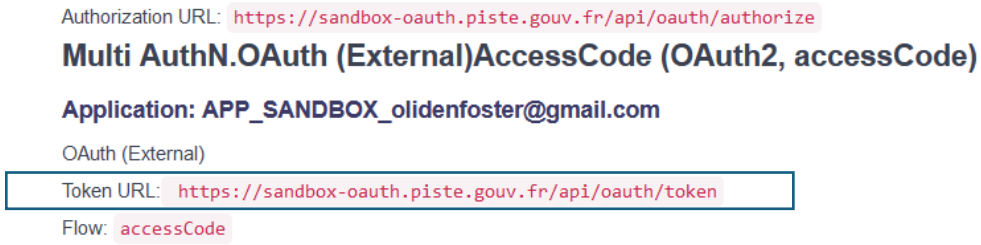
Récupérer ces données ci-dessous :



De plus, dans « API » > « Mes APIs » > « Factures »

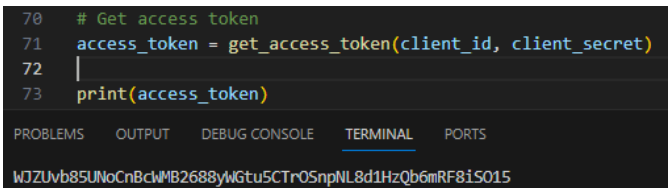


Récupérer ces données ci-dessous :



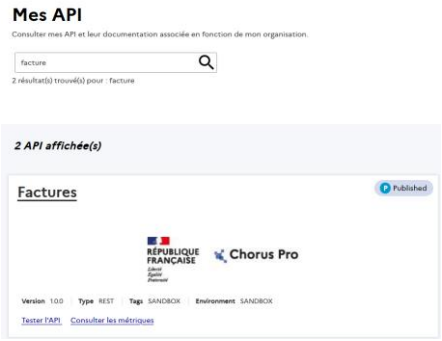
Code python :

```python
# Fonction pour obtenir le jeton d'accès
def getToken(client_id, client_secret,filePath,pathFactureOriginale,pathFichierXML):
    token_url = "https://sandbox-oauth.piste.gouv.fr/api/oauth/token"

    payload = {
        "grant_type": "client_credentials",
        "client_id": client_id,
        "client_secret": client_secret,
        "scope": "openid"
    }

    headers = {
        "Content-Type": "application/x-www-form-urlencoded"
    }

    try:
        # Demande du jeton d'accès à l'URL du jeton
        response = requests.post(token_url, data=payload, headers=headers, verify=True)

        # Lever une HTTPError pour les réponses incorrectes
        response.raise_for_status()

        # Extraction du jeton d'accès du JSON de réponse
        access_token = response.json().get("access_token")
        return access_token
    except requests.exceptions.RequestException as e:
        destination_path = os.path.join(errorPath, os.path.basename(pathFactureOriginale))
        shutil.move(pathFactureOriginale, destination_path)
        os.remove(filePath)
        os.remove(pathFichierXML)
        error_message = f"err {filePath}: \n Erreur dans la demande de jeton : {e}"
        print(error_message)
        logErr.error(error_message)
        log.error(error_message)
        return False
```



b. Envoyé le PDF à chorus pro

Dans « API » > « Mes APIs » > « Factures »



Il n'est plus nécessaire de mettre le paramètre `idUtilisateur`

Tableau d'entrée

Attribut	Typage (tagger)	Format	Règles de gestion	Service fournissant la donnée	Cardinalité
idUtilisateurCourant	Nombre : identifiant technique de l'utilisateur courant dans le système CPP	integer	Cet identifiant permet d'identifier de manière unique l'utilisateur au sein du système CPP 2017		0-1
fichierFlux	String : fichier correspondant au flux de factures, encodé en base64	file			1
nomFichier	String : nom du fichier avec l'extension	varchar(200)			1
syntaxeFlux	String : choix de la syntaxe du fichier à déposer	varchar(24)	Enumération avec 6 valeurs possibles : - IN_DP_E1_UBL_INVOICE - IN_DP_E1_CII_16B - IN_DP_E2_UBL_INVOICE_MIN - IN_DP_E2_CPP_FACTURE_MIN - IN_DP_E2_CII_MIN_16B - IN_DP_E2_CII_FACTURX	recupererSyntaxeFlux	1

https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwiZub7Xg6uEAxWDU6QEHcfMCg8QFnoECBMQAQ&url=https%3A%2F%2Fcommunaute.chorus-pro.gouv.fr%2Fwp-content%2Fuploads%2F2020%2F04%2FSpecifications_Externes_Annexe_Services_API_V5.00.pdf&usg=AOvVaw0lTyQrZ-69rGN6Kmtauvai&opi=89978449

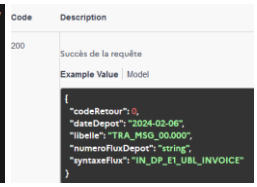
Request URL :

`https://sandbox-api.piste.gouv.fr/cpro/factures/v1/deposer/flux`

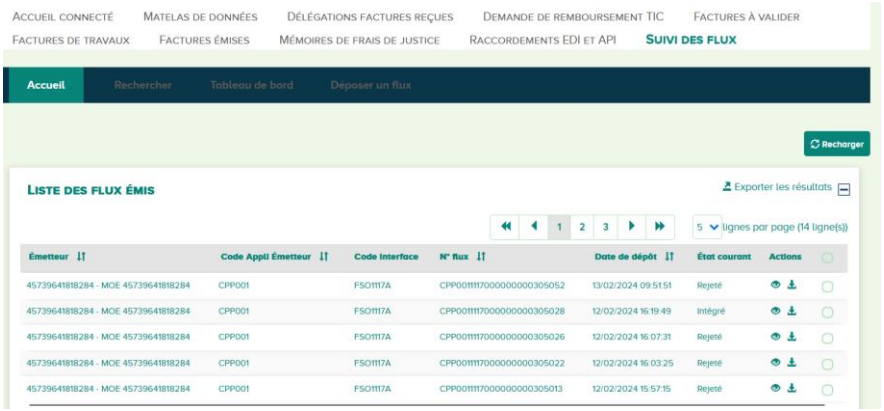
Code python :

```python
# Fonction pour appeler l'API
def sendFact(token, apiUrl, loginTechnique, password, filePath,
file_name,pathFactureOriginale,pathFichierXML):
    try:
        with open(filePath, "rb") as pdf_file:
            # Encoder le contenu du fichier PDF en base64
            encoded_string = base64.b64encode(pdf_file.read()).decode('utf-8')

        headers = {
            "Content-Type": "application/json; charset=UTF-8",
            "Authorization": "Bearer " + token,
            "cpro-account":
base64.b64encode(f"{loginTechnique}:{password}".encode()).decode()
        }

        # Construction de la charge utile JSON
        body_json = {
            "fichierFlux": encoded_string,
            "nomFichier": file_name,  # <-- Le nom du fichier et pas le path
            "syntaxeFlux": "IN_DP_E2_CII_FACTURX",
            "avecSignature": False
        }

        # Effectuer une requête POST à l'API
        response = requests.post(apiUrl, headers=headers,
data=json.dumps(body_json),  verify=True)

        # Lever une HTTPError pour les réponses incorrectes
        response.raise_for_status()

        # Afficher la réponse de l'API
        print(f"{file_name} \n Réponse de l'API :", response.text)
        log.info(f"{file_name} \n Réponse de l'API : {response.text}")
        return json.loads(response.text)
    except requests.exceptions.RequestException as e:
        destination_path = os.path.join(errorPath, os.path.basename(pathFactureOriginale))
        shutil.move(pathFactureOriginale, destination_path)
        os.remove(filePath)
        os.remove(pathFichierXML)
        error_message = f"err {file_name} \n Erreur dans l'envoie du fichier: {e}"
        print(error_message)
        logErr.error(error_message)
        log.error(error_message)
        return None
```

```
PS C:\temp\factures> & C:/Users/ofoster/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/temp/factures/test2.py
Script exécuté le : 2024-02-06 11:36:15
Réponse de l'API : {
  "codeRetour" : 0,
  "libelle" : "GCU_MSG_01_000",
  "numeroFluxDepot" : "CPP00111170000000000304817",
  "dateDepot" : "2024-02-06",
  "syntaxeFlux" : "IN_DP_E2_CII_FACTURX"
}
```



Sur chorus pro : Facturation > Facture émises > suivi des flux > accueil

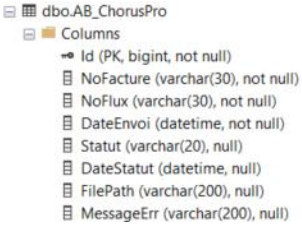


Pour que le code fonctionne en production, il faut changer les liens :

- https://sandbox-oauth.piste.gouv.fr/api/oauth/token -> https://oauth.piste.gouv.fr/api/oauth/token
- https://sandbox-api.piste.gouv.fr/cpro/factures/v1/deposer/flux -> https://api.piste.gouv.fr/cpro/factures/v1/deposer/flux

c. Stockées les factures envoyées dans la base de données

J'ai créé la table AB_ChorusPro :

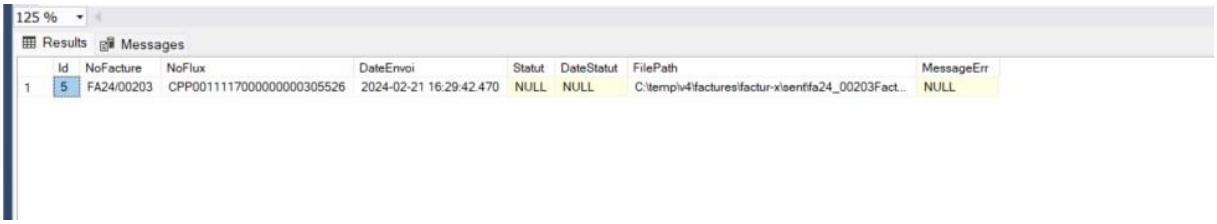


Pour récupérer le numéro de la facture :

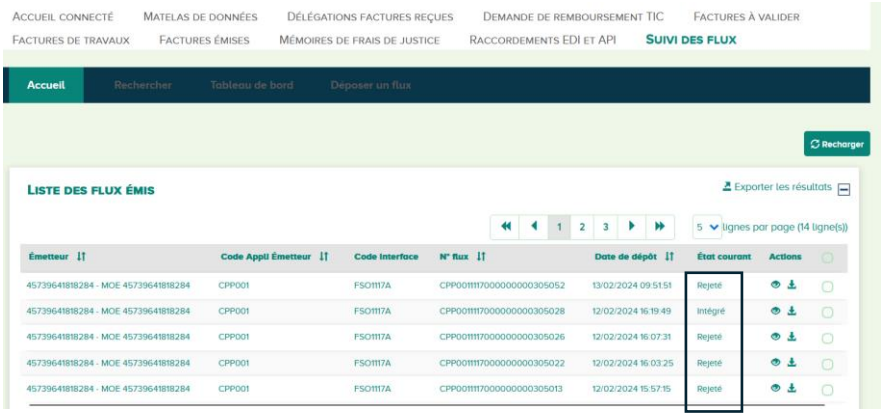
```
def getNoFacture(pdf_path):
    text = ""
    with open(pdf_path, 'rb') as file:
        reader = PyPDF2.PdfReader(file)
        for i in range(len(reader.pages)):
            page = reader.pages[i]
            text += page.extract_text()
    if re.findall(r"FA\d{2}/\d{5}",text):
        return re.findall(r"FA\d{2}/\d{5}",text)[0].replace(' ','')
```

```
def main():
    # Pour chaque fichier PDF dans le dossier spécifié
    for filePath in glob.glob(os.path.join(folder_path, "*.pdf")):
        # Obtenir le nom du fichier PDF
        nomFactureX = str(os.path.basename(filePath))
        noFacture = getNoFacture(filePath)
        if noFacture:
            # Construire les chemins vers les fichiers original et XML
            pathFactureOriginale = os.path.join(fichierOriginal,
str(os.path.basename(filePath)).replace("FactX",""))
            pathFichierXML = os.path.join(fichierOriginal,
str(os.path.basename(filePath)).replace("FactX.pdf",".xml"))
            # Obtenir le jeton d'accès
            token = getToken(client_id,client_secret,filePath)
            if token:
                # Envoyer le fichier à l'API
                api =
sendFact(token,urlDeposerFlux,loginTechnique,password,filePath,nomFactureX,pathFactureOriginale,pathFic
hierXML)
                if api:
                    # Déplacer le fichier vers le dossier "sent"
                    dest_path = os.path.join(folder_path, "sent", nomFactureX)
                    shutil.move(filePath, dest_path)
                    # Obtenir le numéro de flux de dépôt
                    numeroFluxDepot = api['numeroFluxDepot']
                    # Obtenir la date de dépôt
                    dateDepot = current_datetime

                    # Insérer les données dans la base de données
                    cursor.execute('''INSERT INTO AB_ChorusPro (noFacture, noFlux, DateEnvoi, FilePath)
                                      VALUES (?, ?, ?, ?)''',
                                   (noFacture, numeroFluxDepot, dateDepot, dest_path))

                    conn.commit()
                    conn.close()
```

Résultat :



4) Récupérer l'état courant



On veut récupérer ces données et la raison du rejet.

On récupère le token :

```python
# Fonction pour obtenir le jeton d'accès
def get_access_token(client_id, client_secret):
    token_url = "https://sandbox-oauth.piste.gouv.fr/api/oauth/token"

    payload = {
        "grant_type": "client_credentials",
        "client_id": client_id,
        "client_secret": client_secret,
        "scope": "openid"
    }

    headers = {
        "Content-Type": "application/x-www-form-urlencoded"
    }

    try:
        # Demande du jeton d'accès à l'URL du jeton
        response = requests.post(token_url, data=payload, headers=headers, verify=True)

        # Lever une HTTPError pour les réponses incorrectes
        response.raise_for_status()

        # Extraction du jeton d'accès du JSON de réponse
        access_token = response.json().get("access_token")
        return access_token
    except requests.exceptions.RequestException as e:
        print(f"Erreur dans la demande de jeton : {e}")
```

On va repérer les factures qui sont rejeté ou ne possède pas d'état courant :

```
def getFacturesSansEtat():
    server = 'PC-02095' #à modifier
    database = 'SILOG' #à modifier
    conn_str = f'DRIVER={{SQL
Server}};SERVER={server};DATABASE={database};Trusted_Connection=yes;'
    conn = pyodbc.connect(conn_str)
    cursor = conn.cursor()
    sql_query = "SELECT * FROM AB_ChorusPro WHERE Statut IS NULL OR Statut Like
'IN_REJETE'"

    try:
        # Execute the SQL query
        cursor.execute(sql_query)

        # Fetch all the rows returned by the query
        rows = cursor.fetchall()

        # Print the rows
        return rows
    except Exception as e:
        print("An error occurred:", e)
    finally:
        # Close the cursor and connection
        cursor.close()
        conn.close()
```

Puis on fait appel à l'api :

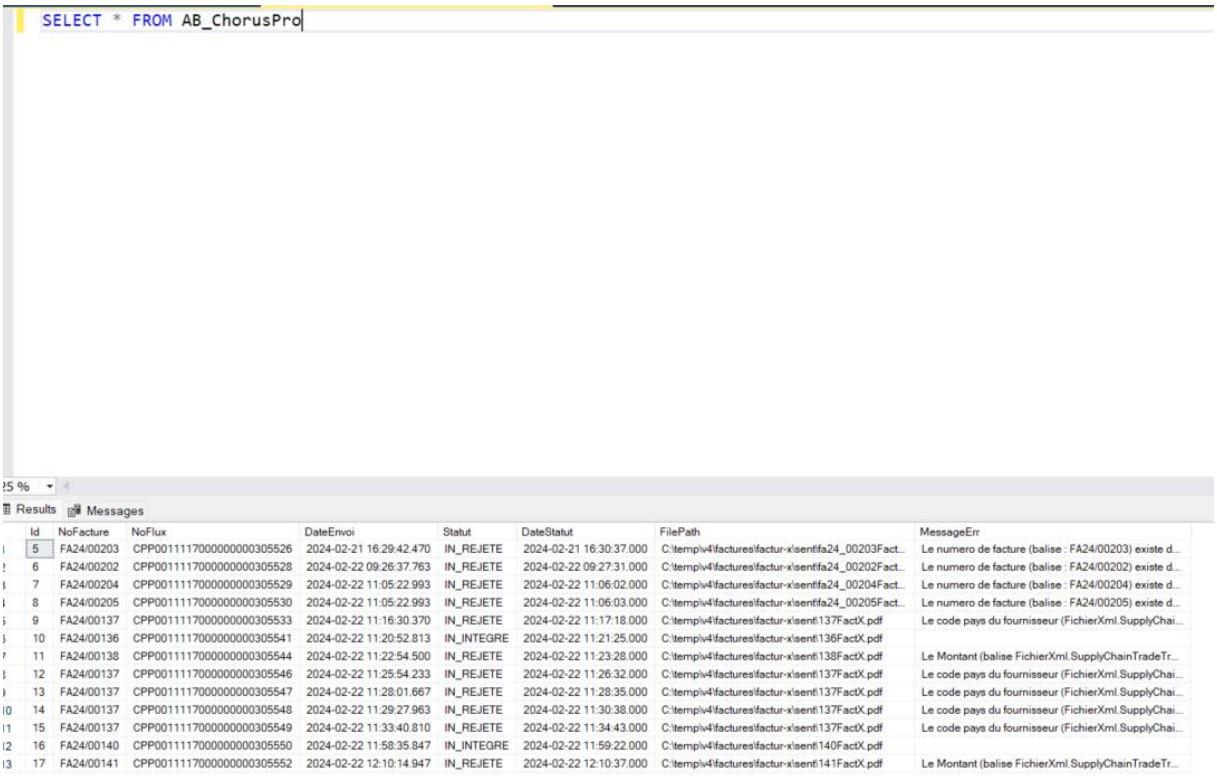
```
https://sandbox-api.piste.gouv.fr/cpro/transverses/v1/consulterCRDetaille
```

(enlevé « -sandbox » pour production)

```
# Fonction pour appeler l'API
def call_api(access_token, api_url, login_technique, password,numeroFluxDepot):

    headers = {
        "Content-Type": "application/json; charset=UTF-8",
        "Authorization": "Bearer " + access_token,
        "cpro-account": base64.b64encode(f"{login_technique}:{password}".encode()).decode()
    }

    # Construction de la charge utile JSON
    body_json = {
        "numeroFluxDepot":numeroFluxDepot,
        "syntaxeFlux": "IN_DP_E2_CII_FACTURX"
    }

    # Effectuer une requête POST à l'API
    response = requests.post(api_url, headers=headers,
data=json.dumps(body_json),  verify=True)
    response_data = json.loads(response.text)
    return response_data
```

Puis on insère les données dans la table :

```
def setData(NoFlux,etatCourant,dateDepot,erreurs):
    server = 'PC-02095' #à modifier
    database = 'SILOG' #à modifier
    conn_str = f'DRIVER={{SQL
Server}};SERVER={server};DATABASE={database};Trusted_Connection=yes;'
    conn = pyodbc.connect(conn_str)
    cursor = conn.cursor()
    update_sql_query = "UPDATE AB_ChorusPro SET Statut = ?,DateStatut = ?, MessageErr = ?
WHERE NoFlux = ?"
    cursor.execute(update_sql_query, (etatCourant,dateDepot, erreurs, NoFlux))

    conn.commit()

    cursor.close()
    conn.close()
```



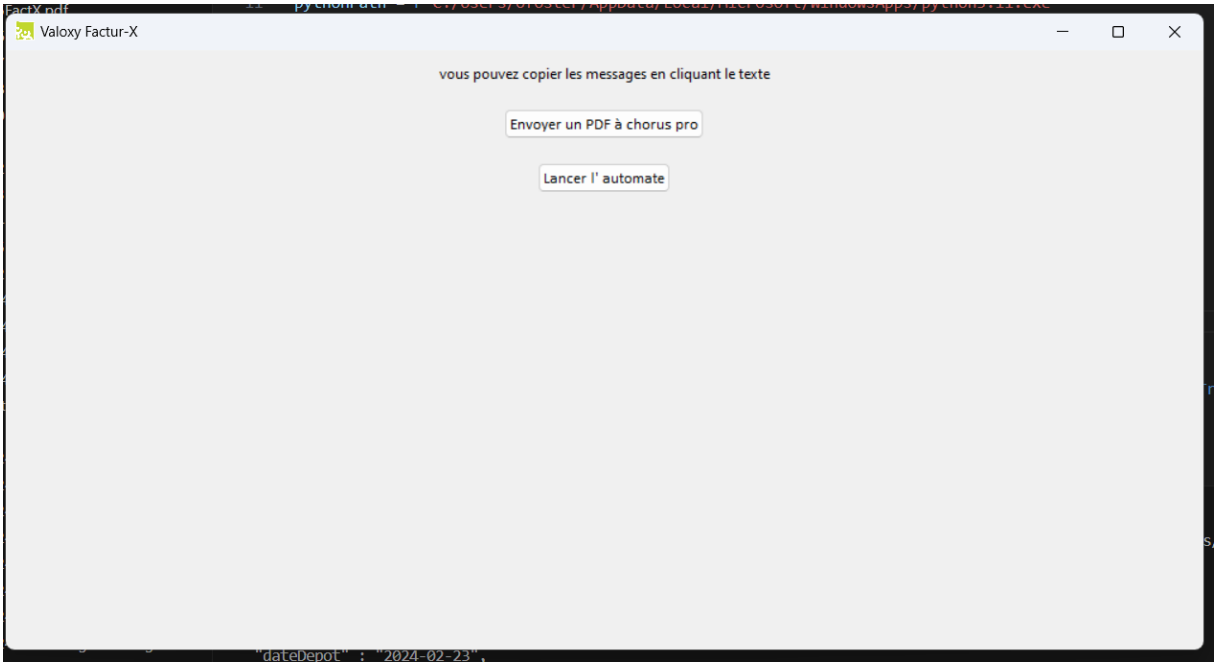
```
SELECT * FROM AB_ChorusPro
```

25 %

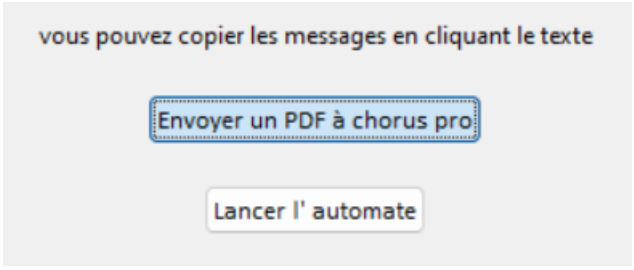
Results | Messages

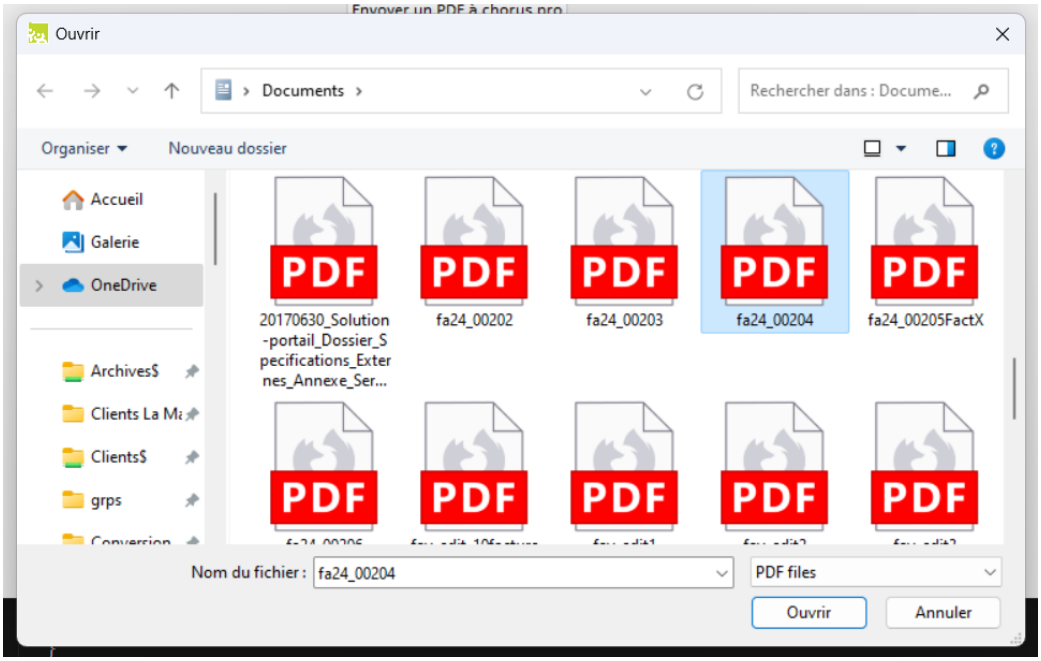
	Id	NoFacture	NoFlux	DateEnvoi	Statut	DateStatut	FilePath	MessageErr
1	5	FA24/00203	CPP00111170000000000305526	2024-02-21 16:29:42.470	IN_REJETE	2024-02-21 16:30:37.000	C:\temp\v4\factures\factur-x\sent\fa24_00203Fact...	Le numero de facture (balise : FA24/00203) existe d...
2	6	FA24/00202	CPP00111170000000000305528	2024-02-22 09:26:37.763	IN_REJETE	2024-02-22 09:27:31.000	C:\temp\v4\factures\factur-x\sent\fa24_00202Fact...	Le numero de facture (balise : FA24/00202) existe d...
3	7	FA24/00204	CPP00111170000000000305529	2024-02-22 11:05:22.993	IN_REJETE	2024-02-22 11:06:02.000	C:\temp\v4\factures\factur-x\sent\fa24_00204Fact...	Le numero de facture (balise : FA24/00204) existe d...
4	8	FA24/00205	CPP00111170000000000305530	2024-02-22 11:05:22.993	IN_REJETE	2024-02-22 11:06:03.000	C:\temp\v4\factures\factur-x\sent\fa24_00205Fact...	Le numero de facture (balise : FA24/00205) existe d...
5	9	FA24/00137	CPP00111170000000000305533	2024-02-22 11:16:30.370	IN_REJETE	2024-02-22 11:17:18.000	C:\temp\v4\factures\factur-x\sent\137FactX.pdf	Le code pays du fournisseur (FichierXml.SupplyChai...
6	10	FA24/00136	CPP00111170000000000305541	2024-02-22 11:20:52.813	IN_INTEGRE	2024-02-22 11:21:25.000	C:\temp\v4\factures\factur-x\sent\136FactX.pdf	
7	11	FA24/00138	CPP00111170000000000305544	2024-02-22 11:22:54.500	IN_REJETE	2024-02-22 11:23:28.000	C:\temp\v4\factures\factur-x\sent\138FactX.pdf	Le Montant (balise FichierXml.SupplyChainTradeTr...
8	12	FA24/00137	CPP00111170000000000305546	2024-02-22 11:25:54.233	IN_REJETE	2024-02-22 11:26:32.000	C:\temp\v4\factures\factur-x\sent\137FactX.pdf	Le code pays du fournisseur (FichierXml.SupplyChai...
9	13	FA24/00137	CPP00111170000000000305547	2024-02-22 11:28:01.667	IN_REJETE	2024-02-22 11:28:35.000	C:\temp\v4\factures\factur-x\sent\137FactX.pdf	Le code pays du fournisseur (FichierXml.SupplyChai...
10	14	FA24/00137	CPP00111170000000000305548	2024-02-22 11:29:27.963	IN_REJETE	2024-02-22 11:30:38.000	C:\temp\v4\factures\factur-x\sent\137FactX.pdf	Le code pays du fournisseur (FichierXml.SupplyChai...
11	15	FA24/00137	CPP00111170000000000305549	2024-02-22 11:33:40.810	IN_REJETE	2024-02-22 11:34:43.000	C:\temp\v4\factures\factur-x\sent\137FactX.pdf	Le code pays du fournisseur (FichierXml.SupplyChai...
12	16	FA24/00140	CPP00111170000000000305550	2024-02-22 11:58:35.847	IN_INTEGRE	2024-02-22 11:59:22.000	C:\temp\v4\factures\factur-x\sent\140FactX.pdf	
13	17	FA24/00141	CPP00111170000000000305552	2024-02-22 12:10:14.947	IN_REJETE	2024-02-22 12:10:37.000	C:\temp\v4\factures\factur-x\sent\141FactX.pdf	Le Montant (balise FichierXml.SupplyChainTradeTr...

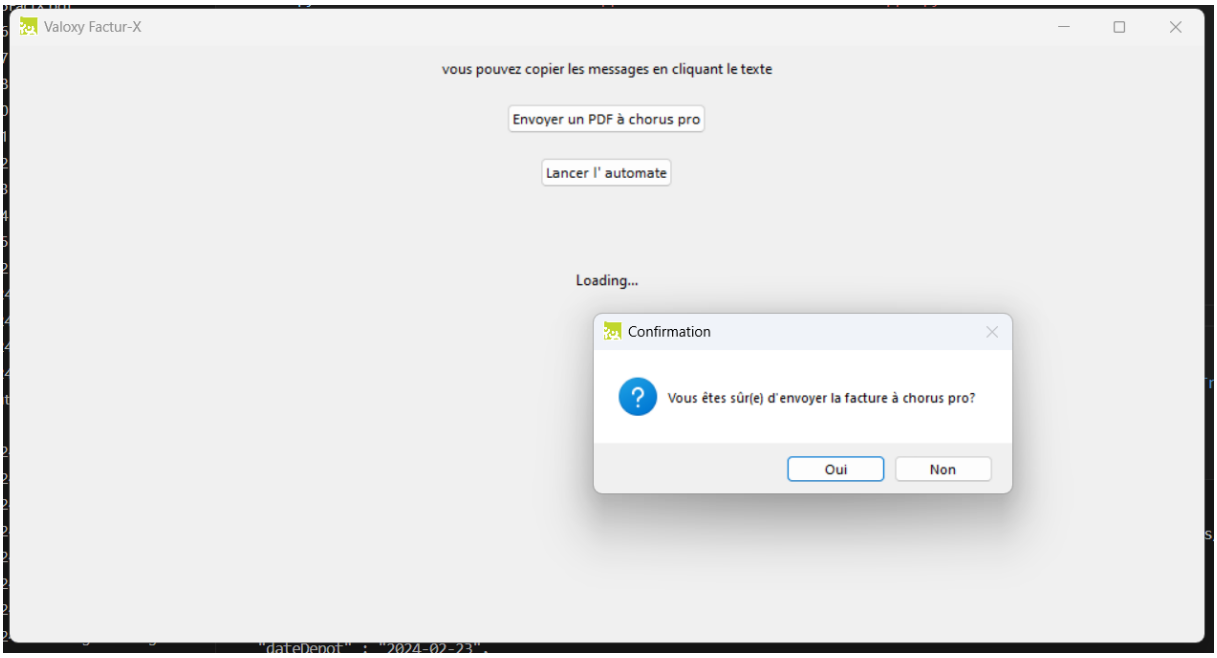
5) Interface

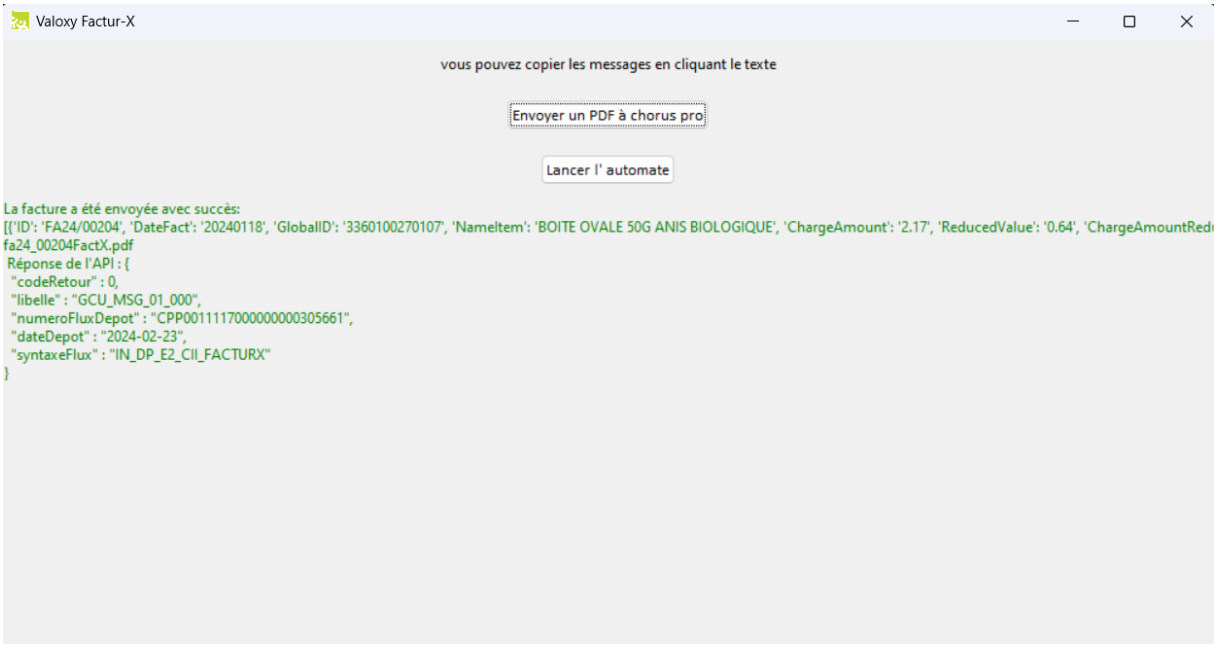


Sélectionner et déposer une facture sur Chorus Pro :









En cas d’erreur :

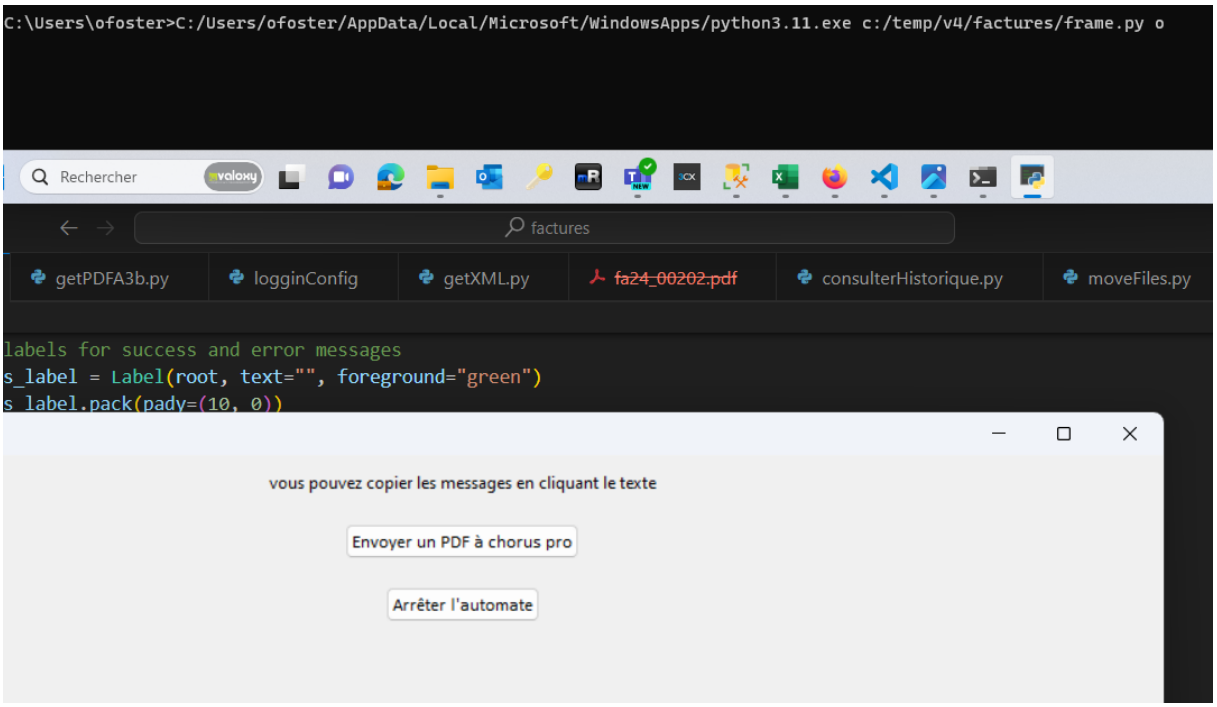
```python
def open_pdf_file():
    error_label.config(text="", foreground="red")
    success_label.config(text="", foreground="green")

    file_path = filedialog.askopenfilename(filetypes=[("PDF files", "*.pdf")])

    if file_path:
        loading_label.config(text="Loading...",foreground="black")
        destination_dir = r"C:\temp\v4\factures"
        file_name = os.path.basename(file_path)
        destination_path = os.path.join(destination_dir, file_name)
        shutil.copy(file_path, destination_path)

        confirmed = messagebox.askyesno("Confirmation", "Vous êtes sûr(e) d'envoyer la
facture à chorus pro?")
        if confirmed:
            try:
                result = subprocess.run(f"{pythonPath} facturX.py", shell=True,
capture_output=True, text=True)

                if result.returncode == 0:
                    if result.stdout.startswith"err"):
                        error_label.config(text=result.stdout, foreground="green")
                    else:
                        success_label.config(text=f"\n{result.stdout}", foreground="green")
                else:
                    error_message = result.stderr.strip() if result.stderr else "Unknown
error occurred."
                    error_label.config(text=error_message, foreground="red")
            except Exception as e:
                error_label.config(text=str(e), foreground="red")
            finally:
                loading_label.config(text=" ",foreground="black")
        else:
            os.remove(destination_path)  # Deleting the copied file
            loading_label.config(text=" ",foreground="black")
    else:
        loading_label.config(text=" ",foreground="black")
```

L’automate:

```
def lancerAutomate():
    global automate_running
    automate_running = not automate_running
    if automate_running:
        automate.config(text="Arrêter l'automate", command=stopAutomate)
        executeAutomate()
    else:
        automate.config(text="Lancer l'automate", command=lancerAutomate)


def executeAutomate():
    if automate_running:
        result = subprocess.run(f"{pythonPath} facturX.py", shell=True,
capture_output=True, text=True)
        # Schedule the function to be called again after ten seconds
        root.after(10000, executeAutomate)


def stopAutomate():
    global automate_running
    automate_running = False
    automate.config(text="Lancer l'automate", command=lancerAutomate)
```

On dépose les fichiers dans le dossier «inputPDFs » qui va effectuer un lancement du programme factur-x chaque 10 secondes

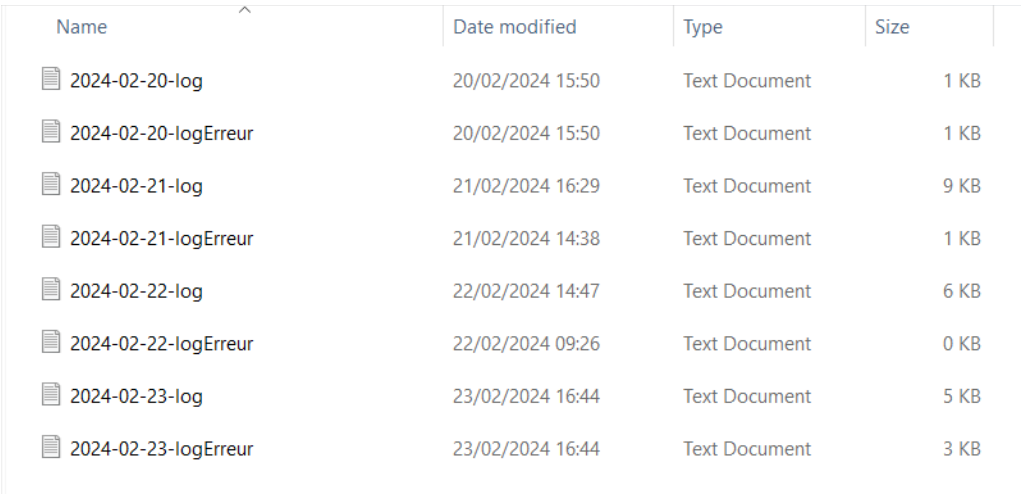
Lancer l’automate par ligne de commande :



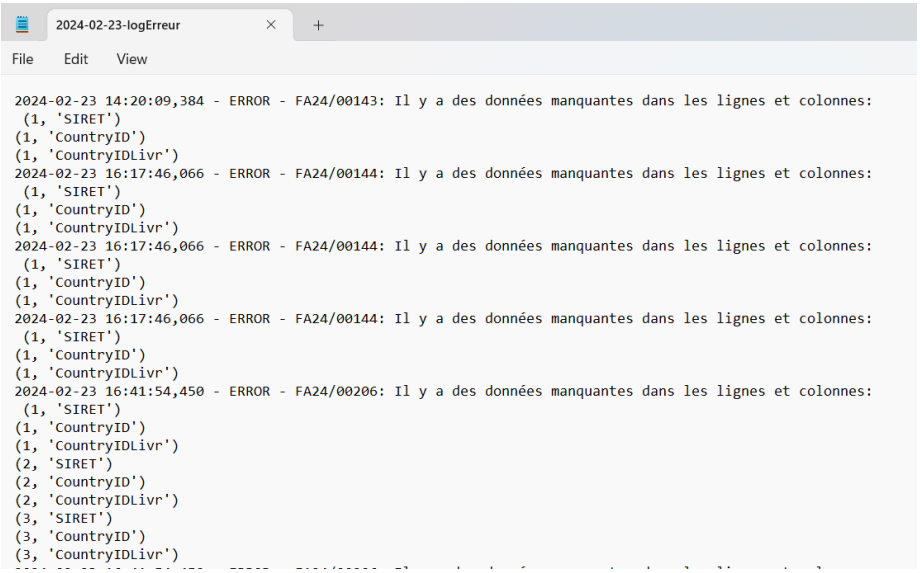

```
if len(sys.argv) > 1 and sys.argv[1] == 'o':
    lancerAutomate()
```

6) Implémentation des logs

On a deux fichiers de logs, un contenant tous les envoie, et un autres contenant que les erreurs.



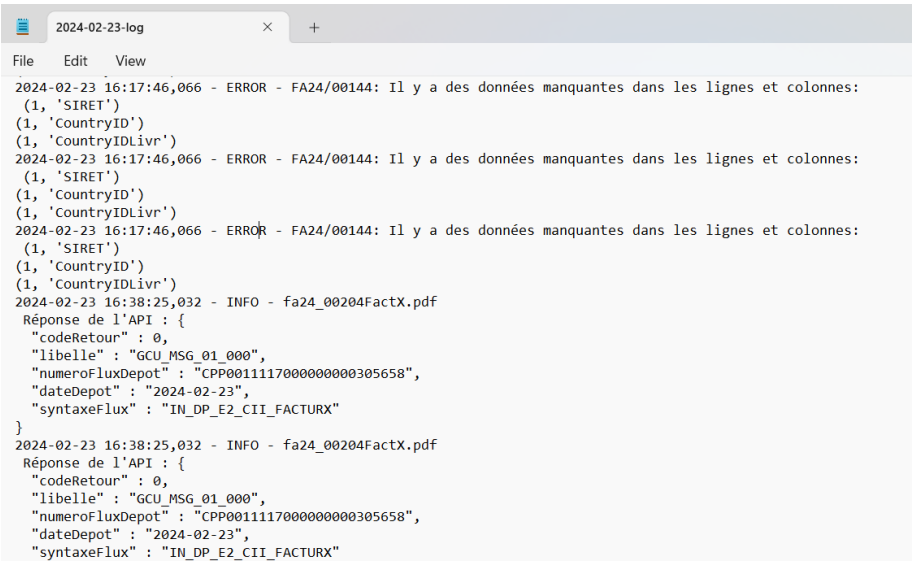
Name	Date modified	Type	Size
2024-02-20-log	20/02/2024 15:50	Text Document	1 KB
2024-02-20-logErreur	20/02/2024 15:50	Text Document	1 KB
2024-02-21-log	21/02/2024 16:29	Text Document	9 KB
2024-02-21-logErreur	21/02/2024 14:38	Text Document	1 KB
2024-02-22-log	22/02/2024 14:47	Text Document	6 KB
2024-02-22-logErreur	22/02/2024 09:26	Text Document	0 KB
2024-02-23-log	23/02/2024 16:44	Text Document	5 KB
2024-02-23-logErreur	23/02/2024 16:44	Text Document	3 KB



2024-02-23-logErreur

File Edit View

```
2024-02-23 14:20:09,384 - ERROR - FA24/00143: Il y a des données manquantes dans les lignes et colonnes:
 (1, 'SIRET')
(1, 'CountryID')
(1, 'CountryIDLivr')
2024-02-23 16:17:46,066 - ERROR - FA24/00144: Il y a des données manquantes dans les lignes et colonnes:
 (1, 'SIRET')
(1, 'CountryID')
(1, 'CountryIDLivr')
2024-02-23 16:17:46,066 - ERROR - FA24/00144: Il y a des données manquantes dans les lignes et colonnes:
 (1, 'SIRET')
(1, 'CountryID')
(1, 'CountryIDLivr')
2024-02-23 16:17:46,066 - ERROR - FA24/00144: Il y a des données manquantes dans les lignes et colonnes:
 (1, 'SIRET')
(1, 'CountryID')
(1, 'CountryIDLivr')
2024-02-23 16:41:54,450 - ERROR - FA24/00206: Il y a des données manquantes dans les lignes et colonnes:
 (1, 'SIRET')
(1, 'CountryID')
(1, 'CountryIDLivr')
(2, 'SIRET')
(2, 'CountryID')
(2, 'CountryIDLivr')
(3, 'SIRET')
(3, 'CountryID')
(3, 'CountryIDLivr')
```



2024-02-23-log

File Edit View

```
2024-02-23 16:17:46,066 - ERROR - FA24/00144: Il y a des données manquantes dans les lignes et colonnes:
 (1, 'SIRET')
(1, 'CountryID')
(1, 'CountryIDLivr')
2024-02-23 16:17:46,066 - ERROR - FA24/00144: Il y a des données manquantes dans les lignes et colonnes:
 (1, 'SIRET')
(1, 'CountryID')
(1, 'CountryIDLivr')
2024-02-23 16:17:46,066 - ERROR - FA24/00144: Il y a des données manquantes dans les lignes et colonnes:
 (1, 'SIRET')
(1, 'CountryID')
(1, 'CountryIDLivr')
2024-02-23 16:38:25,032 - INFO - fa24_00204FactX.pdf
 Réponse de l'API : {
  "codeRetour" : 0,
  "libelle" : "GCU_MSG_01_000",
  "numeroFluxDepot" : "CPP001111700000000305658",
  "dateDepot" : "2024-02-23",
  "syntaxeFlux" : "IN_DP_E2_CII_FACTURX"
}
2024-02-23 16:38:25,032 - INFO - fa24_00204FactX.pdf
 Réponse de l'API : {
  "codeRetour" : 0,
  "libelle" : "GCU_MSG_01_000",
  "numeroFluxDepot" : "CPP001111700000000305658",
  "dateDepot" : "2024-02-23",
  "syntaxeFlux" : "IN_DP_E2_CII_FACTURX"
```

Initialisation des logs :

```
import logging
from datetime import datetime

current_datetime = datetime.now()
# Formater la date et l'heure
formatted_datetime = current_datetime.strftime("%Y-%m-%d")

# Configure logging for the first log file
log = logging.getLogger('log')
log.setLevel(logging.INFO)
log_formatter = logging.Formatter('%(asctime)s - %(levelname)s - %(message)s')
file_handler = logging.FileHandler(fr'C:\temp\v4\factures\logs\{formatted_datetime}-
log.log')
file_handler.setFormatter(log_formatter)
log.addHandler(file_handler)

logErr = logging.getLogger('logErr')
logErr.setLevel(logging.ERROR)
logErr_formatter = logging.Formatter('%(asctime)s - %(levelname)s - %(message)s')
fileErr_handler = logging.FileHandler(fr'C:\temp\v4\factures\logs\{formatted_datetime}-
logErreur.log')
fileErr_handler.setFormatter(logErr_formatter)
logErr.addHandler(fileErr_handler)
```

Ajout des messages dans les logs (exemple getXML.py):

```
# Vérification des données pour s'assurer qu'il n'y a pas de valeurs manquantes
def verifData(data, noFacture,file_path,errorPath):
    vide = []
    for i in range(len(data)):
        for key, val in data[i].items():
            if val == '':
                vide.append((i+1,key))
    if vide:
        error_message = f"{noFacture}: Il y a des données manquantes dans les lignes et
colonnes:\n " + '\n'.join([f"({idx}, '{key}')" for idx, key in vide])
        destination_path = os.path.join(errorPath, os.path.basename(file_path))
        shutil.move(file_path, destination_path)
        print("err " + error_message)
        logErr.error(error_message)
        log.error(error_message)
        return False
    return True
```

(exemple sendFact.py) :

```
        # Afficher la réponse de l'API
        print(f"{file_name} \n Réponse de l'API :", response.text)
        log.info(f"{file_name} \n Réponse de l'API : {response.text}")
```

facturX.py

```
import getPDFs
import getXML
import getPDFA3b
import moveFiles
import sendFact
import getEtatCourant
import time

try:
    getPDFs.main()
    getXML.main()
    getPDFA3b.main()
    moveFiles.main()
    sendFact.main()
    time.sleep(30)
    getEtatCourant.main()
except Exception as e:
    print(f"Error: {e}")
```

7) Changements et évolutions envisageable du projet

- Déposer un fichier contenant plusieurs factures ou un programme qui permet de séparer les factures en fichiers PDF puis de les convertir et de les envoyées
- Implémenter un icone dans la zone de notification
- Une solution qui permet à l'entreprise et les personnes concernées de recevoir un mail contenant les statuts des factures envoyées.

8) Avis sur le projet

J'ai rencontré plusieurs difficultés pendant ce projet, premièrement j'ai dû effectuer des recherches dans les domaines de la juridique et de la comptabilité (format des factures, fonctionnement de Chorus Pro). Deuxièmement, la documentation a été difficile à retrouver et les guides n'était pas à jour. J'ai dû utiliser la plateforme d'assistance de chorus pro et stack et de poser des questions sur Stack Overflow afin d'avancer.

Mais je suis satisfait du travail effectuer malgré mes difficultés rencontrées.

Ce stage m'a permis de me réorienter, j'ai pris l'intérêt de partir dans le secteur scientifique à la place du secteur commerciale de l'informatique.