


a. Stockées les factures envoyées dans la base de données

J'ai créé la table AB_ChorusPro :



Columns
Id (PK, bigint, not null)
NoFacture (varchar(30), not null)
NoFlux (varchar(30), not null)
DateEnvoi (datetime, not null)
Statut (varchar(20), null)
DateStatut (datetime, null)
FilePath (varchar(200), null)
MessageErr (varchar(200), null)

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, pathFichierXML)
                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 :

Results Messages									
	Id	NoFacture	NoFlux	DateEnvoi	Statut	DateStatut	FilePath	MessageErr	
1	5	FA24/00203	CPP0011117000000000305526	2024-02-21 16:29:42.470	NULL	NULL	C:\temp\iv4\factures\factur-x\sentfa24_00203Fact...	NULL	

1) Récupérer l'état courant

ACCUEIL CONNECTÉ MATÉLAS DE DONNÉES DÉLÉGATIONS FACTURES REÇUES DEMANDE DE REMBOURSEMENT TIC FACTURES À VALIDER									
FACTURES DE TRAVAUX FACTURES ÉMISSES MÉMOIRES DE FRAIS DE JUSTICE RACCORDEMENTS EDI ET API SUIVI DES FLUX									
Accueil Rechercher Tableau de bord Deposier un flux									
Recharger									
LISTE DES FLUX ÉMIS Exporter les résultats									
5 lignes par page (14 ligne(s))									
Émetteur	Code Appli Émetteur	Code Interface	N° Flux	Date de dépôt	État courant	Actions			
45739641818284 - MOE 45739641818284	CPP001	FSOMITA	CPP001M7000000000305052	13/02/2024 09:51:54	Rejeté				
45739641818284 - MOE 45739641818284	CPP001	FSOMITA	CPP001M7000000000305028	12/02/2024 16:19:49	Intégré				
45739641818284 - MOE 45739641818284	CPP001	FSOMITA	CPP001M7000000000305026	12/02/2024 16:07:31	Rejeté				
45739641818284 - MOE 45739641818284	CPP001	FSOMITA	CPP001M7000000000305022	12/02/2024 16:03:25	Rejeté				
45739641818284 - MOE 45739641818284	CPP001	FSOMITA	CPP001M7000000000305013	12/02/2024 15:57:15	Rejeté				

On veut récupérer ces données et la raison du rejet.

On récupère le token :

```
# 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

Results								
	Id	NoFacture	NoFlux	DateEnvoi	Statut	DateStatut	FilePath	MessageErr
1	5	FA24/00203	CPP0011117000000000305526	2024-02-21 16:29:42.470	IN_REJETE	2024-02-21 16:30:37.000	C:\temp\4\factures\factur-x\sent\fa24_00203Fact...	Le numero de facture (balise : FA24/00203) existe d...
2	6	FA24/00202	CPP0011117000000000305528	2024-02-22 09:26:37.763	IN_REJETE	2024-02-22 09:27:31.000	C:\temp\4\factures\factur-x\sent\fa24_00202Fact...	Le numero de facture (balise : FA24/00202) existe d...
3	7	FA24/00204	CPP0011117000000000305529	2024-02-22 11:05:22.993	IN_REJETE	2024-02-22 11:06:02.000	C:\temp\4\factures\factur-x\sent\fa24_00204Fact...	Le numero de facture (balise : FA24/00204) existe d...
4	8	FA24/00205	CPP0011117000000000305530	2024-02-22 11:05:22.993	IN_REJETE	2024-02-22 11:06:03.000	C:\temp\4\factures\factur-x\sent\fa24_00205Fact...	Le numero de facture (balise : FA24/00205) existe d...
5	9	FA24/00137	CPP0011117000000000305533	2024-02-22 11:16:30.370	IN_REJETE	2024-02-22 11:17:18.000	C:\temp\4\factures\factur-x\sent\137FactX.pdf	Le code pays du fournisseur (FichierXml SupplyChai...
6	10	FA24/00136	CPP0011117000000000305541	2024-02-22 11:20:52.813	IN_INTEGRE	2024-02-22 11:21:25.000	C:\temp\4\factures\factur-x\sent\136FactX.pdf	
7	11	FA24/00138	CPP0011117000000000305544	2024-02-22 11:22:54.500	IN_REJETE	2024-02-22 11:23:28.000	C:\temp\4\factures\factur-x\sent\138FactX.pdf	Le Montant (balise FichierXml SupplyChainTradeTr...
8	12	FA24/00137	CPP0011117000000000305546	2024-02-22 11:25:54.233	IN_REJETE	2024-02-22 11:26:32.000	C:\temp\4\factures\factur-x\sent\137FactX.pdf	Le code pays du fournisseur (FichierXml SupplyChai...
9	13	FA24/00137	CPP0011117000000000305547	2024-02-22 11:28:01.667	IN_REJETE	2024-02-22 11:28:35.000	C:\temp\4\factures\factur-x\sent\137FactX.pdf	Le code pays du fournisseur (FichierXml SupplyChai...
10	14	FA24/00137	CPP0011117000000000305548	2024-02-22 11:29:27.963	IN_REJETE	2024-02-22 11:30:38.000	C:\temp\4\factures\factur-x\sent\137FactX.pdf	Le code pays du fournisseur (FichierXml SupplyChai...
11	15	FA24/00137	CPP0011117000000000305549	2024-02-22 11:33:40.810	IN_REJETE	2024-02-22 11:34:43.000	C:\temp\4\factures\factur-x\sent\137FactX.pdf	Le code pays du fournisseur (FichierXml SupplyChai...
12	16	FA24/00140	CPP0011117000000000305550	2024-02-22 11:58:35.847	IN_INTEGRE	2024-02-22 11:59:22.000	C:\temp\4\factures\factur-x\sent\140FactX.pdf	
13	17	FA24/00141	CPP0011117000000000305552	2024-02-22 12:10:14.947	IN_REJETE	2024-02-22 12:10:37.000	C:\temp\4\factures\factur-x\sent\141FactX.pdf	Le Montant (balise FichierXml SupplyChainTradeTr...