Python Script for Position Master API

!pip install azure-keyvault

import pandas as pd

import pyodbc

import numpy as np

import json

from azure.identity import ClientSecretCredential

from azure.keyvault.secrets import SecretClient

from azure.identity import ClientSecretCredential

import time

from datetime import datetime, timedelta

today = datetime.today()

yesterday = today - timedelta(days=0)

yesterday=yesterday.strftime('%d-%m-%Y')

yesterday

KEYVAULT\_URI = "https://std-inc-hr-kv-01.vault.azure.net/"

tenant\_id = '1250f2eb-4784-4223-98dc-d6e334122235t'

client\_id = '969c6875-c148-4a09-9d94-66c8f214TRwe2'

client\_secret = '~PX8Q~YzQ\_WjJr1cmd8SH3WpChBAjwUbD5wEr34dO'

credential = ClientSecretCredential(tenant\_id=tenant\_id, client\_id=client\_id, client\_secret=client\_secret)

client = SecretClient(vault\_url=KEYVAULT\_URI, credential=credential)

DEMO\_DB\_PASSWORD=client.get\_secret("STD-INC-HR-SQLDB-01-SK-01").value

server = 'std-inc-hr-sqldbsrv-01.database.windows.net'

db1 = 'STD-INC-HR-SQLDB-01'

username = 'hrsqladmin'

password = DEMO\_DB\_PASSWORD

cnxn = pyodbc.connect('DRIVER={ODBC Driver 18 for SQL Server};SERVER='+server+';DATABASE='+db1+';ENCRYPT=no;UID='+username+';PWD='+ password)

cur = cnxn.cursor()

import requests

url = "https://myjsw.darwinbox.in/orgmasterapi/getpositionMaster"

payload = '{\n    "api\_key": "7f219b02855f424b169e2659580b75c91c26ea3d547f30f7359acaad975f0a687cb3ec80ce9ae99b68bc7107bdf1e733a1defb33e1f88581929d6b54055wEr3",\n    "last\_modified": "%s 12:00:00"\n}'%(yesterday)

headers = {

  'Content-Type': 'text/plain',

  'Authorization': 'Basic anN3X2NvcmVfb25ib2FyZGluZzpQI1FmKDFVZ65ThLq=',

  'Cookie': '\_\_cf\_bm=Q2JpFD5le7x8H1X\_jXeINuu\_yLCsMyCla1LJLYxDjGM-1683557260-0-AcCLZrl4KVJxdscvbdfXJWrpE359d0rcFgfqRPLMNY1OHudj/EhtW24QDKwREnp92qG83P07DMcFRMc=; session=6lvspm08r7sg13gk3nfspdckt2'

}

response = requests.request("POST", url, headers=headers, data=payload)

print(response.text)

data = json.loads(response.text)

a2 = []

for x in data["cols"]:

    a2.append(x)

a1 = []

for x in data["data"]:

    a1.append(x)

df\_PM = pd.DataFrame(a1, columns=a2)

df\_PM.columns=df\_PM.columns.str.replace(' ','\_')

df\_PM.columns=df\_PM.columns.str.replace('-','\_')

df\_PM['Position\_Creation\_Date'] = pd.to\_datetime(df\_PM['Position\_Creation\_Date'],dayfirst=True).dt.strftime('%d-%m-%Y')

df\_PM['Position\_Updation\_Date'] = pd.to\_datetime(df\_PM['Position\_Updation\_Date'],dayfirst=True).dt.strftime('%d-%m-%Y')

df\_PM['Position\_Effective\_Date'] = pd.to\_datetime(df\_PM['Position\_Effective\_Date'],dayfirst=True).dt.strftime('%d-%m-%Y')

df\_PM['Position\_Archived\_effective\_date'] = pd.to\_datetime(df\_PM['Position\_Archived\_effective\_date'],dayfirst=True).dt.strftime('%d-%m-%Y')

df\_PM['Position\_Creation\_Date'] = pd.to\_datetime(df\_PM['Position\_Creation\_Date'],format='%d-%m-%Y')

df\_PM['Position\_Updation\_Date'] = pd.to\_datetime(df\_PM['Position\_Updation\_Date'],format='%d-%m-%Y')

df\_PM['Position\_Effective\_Date'] = pd.to\_datetime(df\_PM['Position\_Effective\_Date'],format='%d-%m-%Y')

df\_PM['Position\_Archived\_effective\_date'] = pd.to\_datetime(df\_PM['Position\_Archived\_effective\_date'],format='%d-%m-%Y')

df\_PM.replace("N.A",np.NaN, inplace = True)

df\_PM.replace(np.NaN," ", inplace = True)

#for df

col = ", ".join([str(i.replace('/','\_').replace('.','')) for i in df\_PM.columns.tolist()])

col

cur.execute("Truncate TABLE Stagging\_Position\_Master;")

cnxn.commit()

#insertion for df1(Active)

for i,row in df\_PM.iterrows():

    sql = "INSERT INTO Stagging\_Position\_Master ("+col+") VALUES (" + "?,"\*(len(row)-1) + "?)"

    cur.execute(sql, tuple(row))

cnxn.commit()

cur.execute("Update\_Production\_Position\_Master;")

cnxn.commit()