import logging

import os

import json

import pyodbc

import pandas as pd

from azure.storage.filedatalake import DataLakeServiceClient

from azure.identity import DefaultAzureCredential

def main(req: func.HttpRequest) -> func.HttpResponse:

logging.info('HTTP trigger function processed a request.')

try:

config = req.get\_json()

file\_system\_name = config.get('file\_system\_name')

file\_path = config.get('file\_path')

file\_type = config.get('file\_type')

database\_config = config.get('database\_config')

if not file\_system\_name or not file\_path or not file\_type or not database\_config:

return func.HttpResponse(

"Missing one or more required parameters.",

status\_code=400

)

file\_content = download\_file\_from\_adls(file\_system\_name, file\_path)

if file\_type.lower() == 'json':

data = process\_json(file\_content)

elif file\_type.lower() == 'parquet':

data = process\_parquet(file\_content)

else:

return func.HttpResponse(

"Unsupported file type.",

status\_code=400

)

insert\_data\_into\_database(data, database\_config)

return func.HttpResponse("Data processed and inserted successfully.", status\_code=200)

except Exception as e:

logging.error(f"Error processing request: {e}")

return func.HttpResponse(f"Error processing request: {e}", status\_code=500)

def download\_file\_from\_adls(file\_system\_name, file\_path):

account\_url = os.environ["ADLS\_ACCOUNT\_URL"]

credential = DefaultAzureCredential()

service\_client = DataLakeServiceClient(account\_url, credential)

file\_system\_client = service\_client.get\_file\_system\_client(file\_system\_name)

file\_client = file\_system\_client.get\_file\_client(file\_path)

download = file\_client.download\_file()

file\_content = download.readall()

return file\_content

def process\_json(file\_content):

return json.loads(file\_content)

def process\_parquet(file\_content):

import io

buffer = io.BytesIO(file\_content)

return pd.read\_parquet(buffer)

def insert\_data\_into\_database(data, database\_config):

conn\_str = database\_config["connection\_string"]

table\_name = database\_config["table\_name"]

conn = pyodbc.connect(conn\_str)

cursor = conn.cursor()

columns = ", ".join(data[0].keys())

values\_placeholder = ", ".join(["?"] \* len(data[0]))

insert\_stmt = f"INSERT INTO {table\_name} ({columns}) VALUES ({values\_placeholder})"

for record in data:

values = tuple(record.values())

cursor.execute(insert\_stmt, values)

conn.commit()

cursor.close()

conn.close()