from flask import Flask, request, jsonify

import os

import base64

import logging

from mbs import process\_excel\_data

app = Flask(\_\_name\_\_)

logger = logging.getLogger(\_\_name\_\_)

DATA\_FOLDER = "input/non\_cricket\_input"

TVR\_OUTPUT\_PATH = "input/TVR Output.xlsx"

SKELETON\_PATH = "input/Skeleton Output.xlsx"

OUTPUT\_PATH = "output/Completed\_Output.xlsx"

def is\_base64\_encoded(s):

try:

return base64.b64encode(base64.b64decode(s)) == s.encode()

except Exception:

return False

@app.route('/process\_pager\_file', methods=['POST'])

def process\_file():

try:

body = request.get\_json()

logger.debug("Request JSON body: %s", body)

file\_info = body if isinstance(body, dict) else None

if not file\_info or file\_info.get("file-type") != "input\_a":

msg = "Missing required file: input\_a"

logger.error(msg)

return jsonify({"error": msg}), 400

filename = file\_info.get("xlsx-name")

attach\_body = file\_info.get("attach-body")

file\_type = file\_info.get("file-type")

logger.info("Processing file: %s of type %s", filename, file\_type)

if not attach\_body or not filename or not file\_type:

msg = f"File '{filename}': Missing 'attach-body', 'xlsx-name', or 'file-type'"

logger.error(msg)

return jsonify({"error": msg}), 400

content = attach\_body.get("contentBytes")

if not content:

msg = f"File '{filename}': Missing 'contentBytes' in attach-body"

logger.error(msg)

return jsonify({"error": msg}), 400

if not is\_base64\_encoded(content):

msg = f"File '{filename}': Invalid base64 content"

logger.error(msg)

return jsonify({"error": msg}), 400

os.makedirs(DATA\_FOLDER, exist\_ok=True)

input\_path = os.path.join(DATA\_FOLDER, filename)

with open(input\_path, "wb") as f:

f.write(base64.b64decode(content))

logger.info("File saved successfully to: %s", input\_path)

try:

# TVR file may be missing, but process\_excel\_data will handle it

process\_excel\_data(

input\_path,

TVR\_OUTPUT\_PATH,

SKELETON\_PATH,

OUTPUT\_PATH

)

if not os.path.exists(OUTPUT\_PATH):

raise FileNotFoundError(f"Output file not created: {OUTPUT\_PATH}")

logger.info(f"File processed successfully. Output saved to {OUTPUT\_PATH}")

with open(OUTPUT\_PATH, "rb") as f:

output\_data = f.read()

encoded\_output = base64.b64encode(output\_data).decode('utf-8')

result = {

"status": "success",

"data": encoded\_output,

"output\_filename": os.path.basename(OUTPUT\_PATH)

}

return jsonify(result)

except Exception as e:

logger.exception("An error occurred while processing the file.")

return jsonify({

"status": "error",

"message": str(e)

}), 500

except Exception as e:

logger.exception("An error occurred while processing the request.")

return jsonify({"error": str(e)}), 500

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)