from flask import Blueprint, request

from flask\_jwt\_extended import jwt\_required, get\_jwt\_identity

from model.model import Bank

from view.view import View

from pymongo import MongoClient

client = MongoClient('mongodb://localhost:27017/')

db = client.bank

bank = Bank(db)

controller\_bp = Blueprint('controller', \_\_name\_\_)

view = View()

@controller\_bp.route('/create\_account', methods=['POST'])

@jwt\_required()

def create\_account():

identity = get\_jwt\_identity()

if identity['role'] != 'banker':

return view.display\_message("Unauthorized", 403)

data = request.json

account\_number = data['account\_number']

owner = data['owner']

if bank.create\_account(account\_number, owner):

return view.display\_message("Account created successfully", 201)

else:

return view.display\_message("Account already exists", 400)

@controller\_bp.route('/deposit', methods=['POST'])

@jwt\_required()

def deposit():

data = request.json

account\_number = data['account\_number']

amount = data['amount']

if bank.deposit(account\_number, amount):

return view.display\_message("Deposit successful")

else:

return view.display\_message("Deposit failed", 400)

@controller\_bp.route('/withdraw', methods=['POST'])

@jwt\_required()

def withdraw():

data = request.json

account\_number = data['account\_number']

amount = data['amount']

if bank.withdraw(account\_number, amount):

return view.display\_message("Withdrawal successful")

else:

return view.display\_message("Withdrawal failed", 400)

@controller\_bp.route('/balance/<account\_number>', methods=['GET'])

@jwt\_required()

def get\_balance(account\_number):

balance = bank.get\_balance(account\_number)

if balance is not None:

return view.display\_balance(balance)

else:

return view.display\_message("Account not found", 404)

@controller\_bp.route('/transactions/<account\_number>', methods=['GET'])

@jwt\_required()

def get\_transaction\_history(account\_number):

transactions = bank.get\_transaction\_history(account\_number)

return view.display\_transactions(transactions)

@controller\_bp.route('/accounts', methods=['GET'])

@jwt\_required()

def get\_all\_accounts():

identity = get\_jwt\_identity()

if identity['role'] != 'banker':

return view.display\_message("Unauthorized", 403)

accounts = bank.get\_all\_accounts()

return view.display\_accounts(accounts)