#Build a REST API with Flask

import json

from flask import Flask, jsonify, request

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

employees = [

{ 'id': 1, 'name': 'Mary' },

{ 'id': 2, 'name': 'Harry' },

{ 'id': 3, 'name': 'Sally' }

]

nextEmployeeId = 4

@app.route('/employees', methods=['GET'])

def get\_employees():

return jsonify(employees)

@app.route('/employees/<int:id>', methods=['GET'])

def get\_employee\_by\_id(id: int):

employee = get\_employee(id)

if employee is None:

return jsonify({ 'error': 'Employee does not exist'}), 404

return jsonify(employee)

def get\_employee(id):

return next((e for e in employees if e['id'] == id), None)

def employee\_is\_valid(employee):

for key in employee.keys():

if key != 'name':

return False

return True

@app.route('/employees', methods=['POST'])

def create\_employee():

global nextEmployeeId

employee = json.loads(request.data)

if not employee\_is\_valid(employee):

return jsonify({ 'error': 'Invalid employee properties.' }), 400

employee['id'] = nextEmployeeId

nextEmployeeId += 1

employees.append(employee)

return '', 201, { 'location': f'/employees/{employee["id"]}' }

@app.route('/employees/<int:id>', methods=['PUT'])

def update\_employee(id: int):

employee = get\_employee(id)

if employee is None:

return jsonify({ 'error': 'Employee does not exist.' }), 404

updated\_employee = json.loads(request.data)

if not employee\_is\_valid(updated\_employee):

return jsonify({ 'error': 'Invalid employee properties.' }), 400

employee.update(updated\_employee)

return jsonify(employee)

@app.route('/employees/<int:id>', methods=['DELETE'])

def delete\_employee(id: int):

global employees

employee = get\_employee(id)

if employee is None:

return jsonify({ 'error': 'Employee does not exist.' }), 404

employees = [e for e in employees if e['id'] != id]

return jsonify(employee), 200

app.run()