Lab 3 Michał Dubiel

from flask import Flask, jsonify, make\_response, request

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

class Pracownik:

    def \_\_init\_\_(self, id, imie, nazwisko, stanowisko):

        self.id = id

        self.imie = imie

        self.nazwisko = nazwisko

        self.stanowisko = stanowisko

    def json(self):

        return jsonify({'id': self.id, 'imie': self.imie, 'nazwisko': self.nazwisko, 'stanowisko': self.stanowisko})

tablica\_pracownikow = [

    Pracownik(1, "Jan", "Kowalski", "Inżynier"),

    Pracownik(2, "Anna", "Nowak", "Księgowa"),

    Pracownik(3, "Marek", "Nowicki", "Programista")

]

def znajdz\_pracownika\_po\_id(id\_szukanego):

    for pracownik in tablica\_pracownikow:

        if pracownik.id == int(id\_szukanego):

            return pracownik

    return 0

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

def get\_pracownik(id):

    pracownik = znajdz\_pracownika\_po\_id(id)

    if pracownik == 0:

        return make\_response(jsonify({'message': 'Pracownik o podanym ID nie istnieje'}), 404)

    else:

        return znajdz\_pracownika\_po\_id(int(id)).json()

@app.route("/workers", methods=["GET"])

def get\_pracownicy():

    return jsonify([{'id': pracownik.id, 'imie': pracownik.imie, 'nazwisko': pracownik.nazwisko, 'stanowisko': pracownik.stanowisko} for pracownik in tablica\_pracownikow])

@app.route("/worker", methods=["POST"])

def dodaj\_pracownika():

    nowy\_pracownik\_id = int(request.json['id'])

    for pracownik in tablica\_pracownikow:

        if pracownik.id == nowy\_pracownik\_id:

            return make\_response(jsonify({'message': 'Pracownik o podanym ID już istnieje'}), 409)

    nowy\_pracownik = Pracownik(id=nowy\_pracownik\_id, imie=request.json['imie'], nazwisko=request.json['nazwisko'], stanowisko=request.json['stanowisko'])

    tablica\_pracownikow.append(nowy\_pracownik)

    return make\_response(jsonify({'message': 'Pracownik został dodany'}), 202)

@app.route("/worker", methods=["PUT"])

def zmien\_pracownika():

    pracownik\_id = int(request.json['id'])

    for pracownik in tablica\_pracownikow:

        if pracownik.id == pracownik\_id:

            pracownik.imie = request.json['imie']

            pracownik.nazwisko = request.json['nazwisko']

            pracownik.stanowisko = request.json['stanowisko']

            return make\_response(jsonify({'message': 'Pracownik o podanym ID został zaaktualizowany'}), 202)

    return make\_response(jsonify({'message': 'Pracownik o podanym ID nie istnieje'}), 404)

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Obraz zawierający zrzut ekranu, tekst, oprogramowanie, komputer

Opis wygenerowany automatycznie

Obraz zawierający zrzut ekranu, tekst, oprogramowanie, komputer

Opis wygenerowany automatycznie

Obraz zawierający zrzut ekranu, tekst, oprogramowanie, komputer

Opis wygenerowany automatycznie

Obraz zawierający zrzut ekranu, tekst, oprogramowanie, komputer

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Obraz zawierający zrzut ekranu, tekst, oprogramowanie, komputer

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

package com.example.restservice;  
import org.springframework.web.client.RestTemplate;  
import org.springframework.http.HttpHeaders;  
import org.springframework.http.ResponseEntity;  
import org.springframework.http.HttpEntity;  
import org.springframework.http.MediaType;  
import org.springframework.http.HttpMethod;  
  
public class WorkerClient {  
  
 private final String baseUrl = "http://127.0.0.1:5000";  
 private final RestTemplate restTemplate;  
 public WorkerClient() {  
 this.restTemplate = new RestTemplate();  
 }  
  
 public String getWorkers() {  
 HttpHeaders headers = new HttpHeaders();  
 headers.setContentType(MediaType.*APPLICATION\_JSON*);  
 HttpEntity<String> requestEntity = new HttpEntity<>(headers);  
 ResponseEntity<String> response = restTemplate.exchange(baseUrl + "/workers",HttpMethod.*GET*,requestEntity, String.class);  
 return response.getBody();  
 }  
  
 public String getWorkerById(String id) {  
 HttpHeaders headers = new HttpHeaders();  
 headers.setContentType(MediaType.*APPLICATION\_JSON*);  
  
 String requestBody = "{\"id\": \"" + id + "\"}";  
 HttpEntity<String> requestEntity = new HttpEntity<>(requestBody, headers);  
  
 ResponseEntity<String> response = restTemplate.exchange(baseUrl + "/worker/2",HttpMethod.*GET*,requestEntity, String.class);  
 return response.getBody();  
 }  
  
 public String PostWorker(String id, String imie, String nazwisko, String stanowisko) {  
 HttpHeaders headers = new HttpHeaders();  
 headers.setContentType(MediaType.*APPLICATION\_JSON*);  
  
 String requestBody = "{\"id\": \"" + id + "\", \"imie\": \"" + imie + "\", \"nazwisko\": \"" + nazwisko + "\", \"stanowisko\": \"" + stanowisko + "\"}";  
 HttpEntity<String> requestEntity = new HttpEntity<>(requestBody, headers);  
  
 ResponseEntity<String> response = restTemplate.exchange(baseUrl + "/worker", HttpMethod.*POST*,requestEntity , String.class);  
 return response.getBody();  
 }  
  
 public String PutWorker(String id, String imie, String nazwisko, String stanowisko) {  
 HttpHeaders headers = new HttpHeaders();  
 headers.setContentType(MediaType.*APPLICATION\_JSON*);  
 String requestBody = "{\"id\": \"" + id + "\", \"imie\": \"" + imie + "\", \"nazwisko\": \"" + nazwisko + "\", \"stanowisko\": \"" + stanowisko + "\"}";  
 HttpEntity<String> requestEntity = new HttpEntity<>(requestBody, headers);  
 ResponseEntity<String> response = restTemplate.exchange(baseUrl + "/worker", HttpMethod.*PUT*, requestEntity, String.class);  
  
 return response.getBody();  
 }  
 public static void main(String[] args) {  
 WorkerClient workerClient = new WorkerClient();  
  
 System.*out*.println("Workers: " + workerClient.getWorkers());  
 System.*out*.println("Worker of ID 2: " + workerClient.getWorkerById("2"));  
 System.*out*.println(workerClient.PostWorker("222", "adam", "małysz", "wiedzmin"));  
 System.*out*.println("Workers: " + workerClient.getWorkers());  
 System.*out*.println(workerClient.PutWorker("222", "adam", "małysz", "Mag 10 lvl"));  
 System.*out*.println("Workers: " + workerClient.getWorkers());  
 }  
}

OUTPUT:

Workers: [{"id":1,"imie":"Jan","nazwisko":"Kowalski","stanowisko":"In\u017cynier"},{"id":2,"imie":"Anna","nazwisko":"Nowak","stanowisko":"Ksi\u0119gowa"},{"id":3,"imie":"Marek","nazwisko":"Nowicki","stanowisko":"Programista"}]

Worker of ID 2: {"id":2,"imie":"Anna","nazwisko":"Nowak","stanowisko":"Ksi\u0119gowa"}

{"message":"Pracownik zosta\u0142 dodany"}

Workers: [{"id":1,"imie":"Jan","nazwisko":"Kowalski","stanowisko":"In\u017cynier"},{"id":2,"imie":"Anna","nazwisko":"Nowak","stanowisko":"Ksi\u0119gowa"},{"id":3,"imie":"Marek","nazwisko":"Nowicki","stanowisko":"Programista"},{"id":222,"imie":"adam","nazwisko":"ma\u0142ysz","stanowisko":"wiedzmin"}]

{"message":"Pracownik o podanym ID zosta\u0142 zaaktualizowany"}

Workers: [{"id":1,"imie":"Jan","nazwisko":"Kowalski","stanowisko":"In\u017cynier"},{"id":2,"imie":"Anna","nazwisko":"Nowak","stanowisko":"Ksi\u0119gowa"},{"id":3,"imie":"Marek","nazwisko":"Nowicki","stanowisko":"Programista"},{"id":222,"imie":"adam","nazwisko":"ma\u0142ysz","stanowisko":"Mag 10 lvl"}]

Process finished with exit code 0

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie