package com.marticus.employeemang.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.marticus.employeemang.entity.EmployeeEntity;

import com.marticus.employeemang.entity.UserEntity;

import com.marticus.employeemang.repo.EmployeeRepo;

import com.marticus.employeemang.repo.UserRepo;

@Service

public class EmployeeService {

@Autowired

UserRepo userRepo;

@Autowired

EmployeeRepo employeeRepo;

public String createUser(UserEntity userEntity) {

String response = " user created successfully";

UserEntity result = userRepo.save(userEntity);

if (result == null) {

response = "User cannot be created ";

return response;

}

return response;

}

public String update(UserEntity userEntity) {

String response = "User updated Successfully";

UserEntity user = userRepo.findUsersById(userEntity.getId());

if (user != null) {

userRepo.save(userEntity);

return response;

}

response = "No user found";

return response;

}

public String deleteUser(int id) {

String response = "User deleted Successfully";

UserEntity user = userRepo.findUsersById(id);

if (user != null) {

userRepo.deleteById(user.getId());

return response;

}

response = "User Not Found";

return response;

}

public String findAll() {

String temp;

String response = "";

List<UserEntity> users = userRepo.findAll();

if (users.size() > 0) {

for (UserEntity user : users) {

temp = "\n Id : " + user.getId() + "\n Name : " + user.getUserName() + "\n";

response = response + temp;

}

return response;

}

response = "There are no users";

return response;

}

public String findUsersById(int id) {

String response;

UserEntity user = userRepo.findUsersById(id);

if (user != null) {

response = " Id : " + user.getId() + "\n Name : " + user.getUserName();

return response;

}

response = "User Not Found";

return response;

}

public String findUsersByUserName(String userName) {

String response;

UserEntity user = userRepo.findUsersByUserName(userName);

if (user != null) {

response = " Id : " + user.getId() + "\n Name : " + user.getUserName();

return response;

}

response = "User Not Found";

return response;

}

public boolean authenticateUser(UserEntity user) {

UserEntity userEntity = userRepo.findUsersByUserNameAndPassword(user.getUserName(), user.getPassword());

if (userEntity != null) {

return true;

}

return false;

}

public List<EmployeeEntity> getAllEmployeeList() {

List<EmployeeEntity> employees = employeeRepo.findAll();

return employees;

}

public void saveEmpInDB(EmployeeEntity employeeEntity) {

employeeRepo.save(employeeEntity);

}

public EmployeeEntity findEmployeesById(int id) {

return employeeRepo.findEmployeesById(id);

}

public void deleteEmp(int id) {

EmployeeEntity employeeEntity = employeeRepo.findEmployeesById(id);

if (employeeEntity != null) {

employeeRepo.delete(employeeEntity);

}

}

public List<EmployeeEntity> search(String search) {

return employeeRepo.findEmployeesByEmpNameContaining(search);

}

public void update(EmployeeEntity employeeEntity) {

EmployeeEntity emp = employeeRepo.findEmployeesById(employeeEntity.getId());

if(emp!=null)

{

emp.setEmpName(employeeEntity.getEmpName());

emp.setAge(employeeEntity.getAge());

emp.setSalary(employeeEntity.getSalary());

emp.setDepartment(employeeEntity.getDepartment());

emp.setAadharNumber(employeeEntity.getAadharNumber());

employeeRepo.save(emp);

}

}

}