

Employee.java

package com.entity;

import static org.hamcrest.CoreMatchers.notNullValue;

import java.util.Date;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Temporal;

import javax.persistence.TemporalType;

import com.fasterxml.jackson.annotation.JsonIgnore;

import com.fasterxml.jackson.annotation.JsonInclude;

import com.fasterxml.jackson.annotation.JsonInclude.Include;

import lombok.Data;

import lombok.Setter;

@Entity

//@Data

//@Setter

@JsonInclude(value = Include.NON\_EMPTY)

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int eid;

private String ename;

private String erole;

private String gender;

private String epass;

@Temporal(TemporalType.DATE)

private Date date;

public int getEid() {

return eid;

}

public void setEid(int eid) {

this.eid = eid;

}

public String getEname() {

return ename;

}

public void setEname(String ename) {

this.ename = ename;

}

public String getErole() {

return erole;

}

public void setErole(String erole) {

this.erole = erole;

}

public String getGender() {

return gender;

}

public void setGender(String gender) {

this.gender = gender;

}

public String getEpass() {

return epass;

}

public void setEpass(String epass) {

this.epass = epass;

}

public Date getDate() {

return date;

}

@JsonIgnore

public void setDate(Date date) {

this.date = date;

}

@Override

public String toString() {

return "Employee [eid=" + eid + ", ename=" + ename + ", erole=" + erole + ", gender=" + gender + ", epass="

+ epass + ", date=" + date + "]";

}

}

Emprepository.java

package com.repository;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.jpa.repository.support.JpaRepositoryFactory;

import org.springframework.stereotype.Repository;

import com.entity.Employee;

@Repository

public interface emprepository extends JpaRepository<Employee, Integer> {

}

employeeDT

**package** com.DTO;

**public** **class** employeeDTO {

**private** String ename;

**private** String erole;

**private** String gender;

@Override

**public** String toString() {

**return** "employeeDTO [ename=" + ename + ", erole=" + erole + ", gender=" + gender + "]";

}

**public** String getEname() {

**return** ename;

}

**public** **void** setEname(String ename) {

**this**.ename = ename;

}

**public** String getErole() {

**return** erole;

}

**public** **void** setErole(String erole) {

**this**.erole = erole;

}

**public** String getGender() {

**return** gender;

}

**public** **void** setGender(String gender) {

**this**.gender = gender;

}

}

ServiceImplementation.java

**package** com.service;

**import** java.util.Date;

**import** java.util.List;

**import** org.slf4j.Logger;

**import** org.slf4j.LoggerFactory;

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

**import** org.springframework.stereotype.Service;

**import** com.entity.Employee;

**import** com.repository.emprepository;

@Service

**public** **class** ServiceImplementation **implements** empservice {

Logger logger = LoggerFactory.*getLogger*(ServiceImplementation.**class**);

@Autowired

**private** emprepository er;

@Override

**public** **int** addempservice(Employee emp) {

**if** (emp != **null**) {

emp.setDate(**new** Date());

er.save(emp);

logger.info("data emp addes" + emp);

} **else** {

logger.warn("null emp not added");

**return** 1;

}

**return** 0;

}

@Override

**public** List<Employee> getuserservice() {

List<Employee> findAllemp = er.findAll();

**return** findAllemp;

}

}

empController.java

**package** com.controller;

**import** java.util.ArrayList;

**import** java.util.List;

**import** org.slf4j.Logger;

**import** org.slf4j.LoggerFactory;

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

**import** org.springframework.http.HttpStatus;

**import** org.springframework.http.ResponseEntity;

**import** org.springframework.web.bind.annotation.GetMapping;

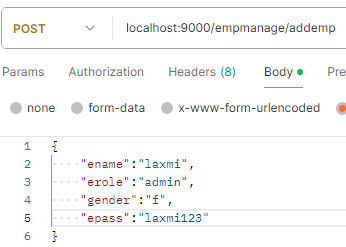
**import** org.springframework.web.bind.annotation.PostMapping;

**import** org.springframework.web.bind.annotation.RequestBody;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestParam;

**import** org.springframework.web.bind.annotation.RestController;



**import** com.DTO.employeeDTO;

**import** com.entity.Employee;

**import** com.service.empservice;

@RestController

@RequestMapping("/empmanage")

**public** **class** empController {

@Autowired

**private** empservice es;

Logger logger = LoggerFactory.*getLogger*(empController.**class**);

@PostMapping("/addemp")

**public** ResponseEntity<?> addemp(@RequestBody Employee emp) {

logger.info("in add emp" + emp);

**int** result = es.addempservice(emp);

**if** (result == 0) {

**return** **new** ResponseEntity("user added succesfully", HttpStatus.***OK***);

} **else** {

**return** **new** ResponseEntity("user is null", HttpStatus.***NOT\_ACCEPTABLE***);

}

}

@GetMapping("/loginadmin")

**public** ResponseEntity<?> loginadmin(@RequestParam("erole") String erole, @RequestParam("ename") String enam,

@RequestParam("epass") String epass) {

logger.info("admin access controller");

**return** **new** ResponseEntity("user added succesfully", HttpStatus.***OK***);

}

@GetMapping("/getalluser")

**public** ResponseEntity<?> getalluser()

{

List<Employee> list = es.getuserservice();

 **return** **new** ResponseEntity(list, HttpStatus.***OK***);

 }

@GetMapping("/getalluserexcept")

**public** ResponseEntity<?> getalluserexcept()

{

List<Employee> listexcept = es.getuserservice();

List<employeeDTO> edtolist= **new** ArrayList<employeeDTO>();

**for**(Employee emp:listexcept)

 {

employeeDTO edto=**new** employeeDTO();

edto.setEname(emp.getEname());

edto.setErole(emp.getErole());

edto.setGender(emp.getGender());

edtolist.add(edto);

}

**return** **new** ResponseEntity(edtolist, HttpStatus.***OK***);

}

}

@GetMapping("/getRoleEmailOnlybyID")

**public** ResponseEntity<?> getRoleEmailOnlybyID(@RequestParam ("eid")**int** id)

{

Employee emp = es.getRoleEmailOnlybyIDserv(id);

RoleGenderbyIdDTO dto2=**new** RoleGenderbyIdDTO();

dto2.setErole(emp.getErole());

dto2.setGender(emp.getGender());

**return** **new** ResponseEntity(dto2, HttpStatus.***OK***);

}

**public** **class** RoleGenderbyIdDTO {

**private** String erole;

**private** String gender;

**public** String getErole() {

**return** erole;

}

**public** **void** setErole(String erole) {

**this**.erole = erole;

}

**public** String getGender() {

**return** gender;

}

**public** **void** setGender(String gender) {

**this**.gender = gender;

}

@Override

**public** String toString() {

**return** "RoleGenderbyIdDTO [erole=" + erole + ", gender=" + gender + "]";

}

}

@Override

**public** Employee getRoleEmailOnlybyIDserv(**int** id) {

Employee empById =er.findbyidonly(id);

**return** empById;

}