**Postman mysql**

**Controller.java**

package com.example.demo.controller;

import java.util.List;

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

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

import org.springframework.web.bind.annotation.PathVariable;

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

import org.springframework.web.bind.annotation.PutMapping;

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

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

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

import com.example.demo.model.Model;

import com.example.demo.repository.Repo;

@RestController

@RequestMapping("/users")

public class Controller {

@Autowired

private Repo repo;

@GetMapping

public List<Model> getAllUser()

{

return this.repo.findAll();

}

@GetMapping("/{id}")

public Model getUserById(@PathVariable(value="id") long userId) {

return this.repo.findById(userId)

.orElseThrow();

}

@PostMapping

public Model createUser(@RequestBody Model model)

{

return this.repo.save(model);

}

@PutMapping("/{id}")

public Model updateUser(@RequestBody Model model,@PathVariable("id") long userId)

{

//return this.userRepository.save(user);

Model ex=this.repo.findById(userId)

.orElseThrow();

ex.setFirstname(model.getFirstname());

ex.setLasttname(model.getLasttname());

return this.repo.save(ex);

}

}

**Model.java**

package com.example.demo.model;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

@Entity

@Table(name="user")

public class Model {

@Id

@GeneratedValue(strategy=GenerationType.AUTO)

private Long id;

private String firstname;

private String lasttname;

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getFirstname() {

return firstname;

}

public void setFirstname(String firstname) {

this.firstname = firstname;

}

public String getLasttname() {

return lasttname;

}

public void setLasttname(String lasttname) {

this.lasttname = lasttname;

}

public Model(Long id, String firstname, String lasttname)

{

super();

this.id = id;

this.firstname = firstname;

this.lasttname = lasttname;

}

public Model()

{

}

}

**Repo.java**

package com.example.demo.repository;

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

import org.springframework.stereotype.Repository;

import com.example.demo.model.Model;

@Repository

public interface Repo extends JpaRepository<Model, Long>{

}

**Application properties**

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.url = jdbc:mysql://localhost:3306/my-sql

spring.datasource.username=root

spring.datasource.password=Tiger@123

spring.jpa.hibernate.ddl-auto = update

spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5Dialect

**crud operations mysql**

**Econtroller.java**

package com.example.empp.controller;

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

import org.springframework.stereotype.Controller;

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

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

import org.springframework.web.servlet.ModelAndView;

import com.example.empp.model.Edemo;

import com.example.empp.repository.Emrepo;

@Controller

public class Econtroller {

@Autowired

Emrepo emrepo;

@RequestMapping("index")

public String index()

{

return "index.jsp";

}

@RequestMapping("addEmp")

public String addEmp(Edemo edemo)

{

emrepo.save(edemo);

return "index.jsp";

}

@RequestMapping("getEmp")

public ModelAndView getEmp(@RequestParam int id)

{

ModelAndView mv=new ModelAndView("disemply.jsp");

Edemo edemo=emrepo.findById(id).orElse(new Edemo());

mv.addObject(edemo);

return mv;

}

@RequestMapping("deleteEmp")

public ModelAndView deleteEmp(@RequestParam int id)

{

ModelAndView mv = new ModelAndView("deletedDisplay.jsp");

Edemo edemo=emrepo.findById(id).orElse(new Edemo());

emrepo.deleteById(id);

mv.addObject(edemo);

return mv;

}

@RequestMapping("UpdateEmp")

public ModelAndView updateEmp(Edemo edemo)

{

ModelAndView mv=new ModelAndView("UpdateEmployee.jsp");

edemo=emrepo.findById(edemo.getId()).orElse(new Edemo());

mv.addObject(edemo);

return mv;

}

}

**Edemo.java**

package com.example.empp.model;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

@Entity

public class Edemo {

@Id

private int id;

private String name;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

@Override

public String toString() {

return "Edemo [id=" + id + ", name=" + name + "]";

}

}

**Emrepo.java**

package com.example.empp.repository;

import org.springframework.data.repository.CrudRepository;

import com.example.empp.model.Edemo;

public interface Emrepo extends CrudRepository<Edemo,Integer>{

}

**Application properties**

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.url = jdbc:mysql://localhost:3306/krdb

spring.datasource.username=root

spring.datasource.password=Tiger@123

spring.jpa.hibernate.ddl-auto = update

spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5Dialect

**deletedDisplay.jsp**

<html>

<head>

<title></title>

</head>

<body>

<h2>Deleted Employee</h2>

${demo}

Name:${edemo.name}

<h2>Deleted Successfully</h2>

</body>

</html>

**Disemply.jsp**

Id:${edemo.id}

Name:${edemo.name}

**Index.jsp**

<html>

<head>

<title></title>

</head>

<body>

<h2>Add Employee</h2>

<form action=*"addEmp"*>

EID:<input type=*"text"* name=*"id"*><br><br>

<input type=*"submit"*>

</form>

<h2>Display Employee</h2>

<form action=*"getEmp"*>

EID:<input type=*"text"* name=*"id"*><br><br>

<input type=*"submit"*>

</form>

<h2>Delete Employee</h2>

<form action=*"deleteEmp"*>

EID:<input type=*"text"* name=*"id"*><br><br>

<input type=*"submit"*>

</form>

<h2>Update Employee</h2>

<form action=*"UpdateEmp"*>

EID:<input type=*"text"* name=*"id"*><br><br>

<input type=*"submit"*>

</form>

</body>

</html>

**UpdateEmployee.jssp**

<html>

<head>

<title></title>

</head>

<body>

<h2>To be Update Employee:${edemo.id}</h2>

<form action=*"addEmp"*>

EID:<input type=*"text"* name=*"id"* value=*"*${edemo.id}*"* readonly=*"readonly"*><br>

EName:<input type=*"text"* name=*"name"* value=*"*${edemo.name}*"*><br>

<input type=*"submit"*>

</form>

</body>

</html>

**Mongodb studentreg**

**Stucontroller**

package com.info.studentreg.controller;

import java.util.List;

import java.util.Optional;

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

import org.springframework.web.bind.annotation.DeleteMapping;

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

import org.springframework.web.bind.annotation.PathVariable;

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

import org.springframework.web.bind.annotation.PutMapping;

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

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

import com.info.studentreg.model.Student;

import com.info.studentreg.repository.Sturepo;

@RestController

public class Stucontroller {

@Autowired

Sturepo sturepo;

@PostMapping("/addStu")

public String addStu(@RequestBody Student stu) {

sturepo.save(stu);

return "Record inserted"+stu.getId();

}

@GetMapping("/display")

public List<Student> display(){

return sturepo.findAll();

}

@GetMapping("/search/{id}")

public Optional<Student> search(@PathVariable int id){

return sturepo.findById(id);

}

@DeleteMapping("/delete/{id}")

public String delete(@PathVariable int id) {

sturepo.deleteById(id);

return "Record deletedby" + id;

}

@PutMapping("/update")

public String update(@RequestBody Student stu) {

Student existstu=sturepo.findById(stu.getId()).get();

existstu.setName(stu.getName());

existstu.setBranch(stu.getBranch());

sturepo.save(existstu);

return "Record updated bi"+stu.getId();

}

}

**Student.java**

package com.info.studentreg.model;

import org.springframework.data.annotation.Id;

import org.springframework.data.mongodb.core.mapping.Document;

@Document("stureg")

public class Student {

@Id

private int id;

private String name;

private String branch;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getBranch() {

return branch;

}

public void setBranch(String branch) {

this.branch = branch;

}

@Override

public String toString() {

return "Student [id=" + id + ", name=" + name + ", branch=" + branch + "]";

}

}

**Sturepo.java**

package com.info.studentreg.repository;

import org.springframework.data.mongodb.repository.MongoRepository;

import com.info.studentreg.model.Student;

public interface Sturepo extends MongoRepository<Student, Integer>{

}

**Application properties**

spring.data.mongodb.host=localhost

spring.data.mongodb.port=27017

spring.data.mongodb.database=studentdb

mongodb+srv://admin:<password>@cluster0.yjtftp2.mongodb.net/test