DAY-9

Spbackendapp.java

package Spring;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import io.swagger.v3.oas.annotations.OpenAPIDefinition;

@SpringBootApplication

//Swagger UI Header Configuration

@OpenAPIDefinition(

info =@Info(

// Swagger Documentation Title

title = "SpringBoot - Backend - IRC - Sample - NEO",

// API Version info

version = "1.0.2",

description = "Sample IRC Project",

contact = @Contact(

// API Author

name = "MaX",

// Author e-mail

email = "max.neo.dev@gmail.com"

)

)

)

public class Spbackendapp {

public static void main(String[] args) {

SpringApplication.run(Spbackendapp.class, args);

}

}

SERVICE.JAVA

package Service;

import java.util.List;

import java.util.Optional;

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

import org.springframework.stereotype.Service;

import Model.model;

import Model.amodel;

import Repo.arepo;

import Repo.aurepo;

@Service

public class service {

@Autowired

private arepo apprepo;

@Autowired

private aurepo authrepo;

//Login Logic

public String Login(String username, String password) {

amodel xuser = authrepo.findByusername(username);

if (xuser == null) {

return "invalidusername";

} else {

if (xuser.getPassword().equals(password)) {

return "success";

} else {

return "invalidpassword";

}

}

}

//Signup Logic

public String Signup(amodel xuser) {

String username = xuser.getUsername();

amodel authuser = authrepo.findByusername(username);

if (authuser == null) {

authrepo.save(xuser);

return "useradded";

} else {

return "existingusername";

}

}

// Get

public List<amodel> getData() {

return arepo.findAll();

}

// Post

public model addData(model data) {

return arepo.save(data);

}

// Edit (modify if data exists and save, else don't save)

public model editData(model data, Long id) {

model edx = apprepo.findById(id).orElse(data);

if (edx != null) {

edx.setProductname(data.getProductname());

edx.setProductprice(data.getProductprice());

edx.setProductrating(data.getProductrating());

edx.setProductstock(data.getProductstock());

edx.setProductimg(data.getProductimg());

return apprepo.saveAndFlush(edx);

} else {

return null;

}

}

//Delete

public String deleteData(Long id) {

apprepo.deleteById(id);

return "Deleted Successfully";

}

//Find by ID

public Optional<model> findbyID(Long id) {

return arepo.findById(id);

}

}