Spring MVC

Web.xml :

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"* id=*"WebApp\_ID"* version=*"3.1"*>

<display-name>SpringMVC1</display-name>

<servlet>

<servlet-name>jd</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<!-- Map all requests to the DispatcherServlet for handling -->

<servlet-mapping>

<servlet-name>jd</servlet-name>

<url-pattern>\*.htm</url-pattern>

</servlet-mapping>

</web-app>

Jd-servlet.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd"*>

<bean id=*"/welcome.htm"* class=*"com.jd.controller.WelcomeController"*></bean>

<bean id=*"viewResolver"* class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

<property name=*"prefix"* value=*"/pages/"*></property>

<property name=*"suffix"* value=*".jsp"*></property>

</bean>

</beans>

Folder under Webcontent/pages

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title></title>

</head>

<body>

<h2>${msg}</h2>

</body>

</html>

WelcomeController .java “

package com.jd.controller;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.springframework.web.servlet.ModelAndView;

import org.springframework.web.servlet.mvc.Controller;

public class WelcomeController implements Controller {

@Override

public ModelAndView handleRequest(HttpServletRequest arg0, HttpServletResponse arg1) throws Exception {

System.out.println("inside invoking the business layer.");

ModelAndView mav = new ModelAndView();

mav.addObject("msg", "Welcome to Spring MVC");

mav.setViewName("hello"); // logical view name.

return mav;

}

}

Second Example with annotation :

Web.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"* id=*"WebApp\_ID"* version=*"3.1"*>

<display-name>EmployeeManagement</display-name>

<!-- The front controller of this Spring Web application, responsible for handling all application requests -->

<servlet>

<servlet-name>jd</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<!-- Map all requests to the DispatcherServlet for handling -->

<servlet-mapping>

<servlet-name>jd</servlet-name>

<url-pattern>\*.htm</url-pattern>

</servlet-mapping>

</web-app>

Empform.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title></title>

</head>

<body bgcolor=*"cyan"* align=*"center"*>

<form id=*"form1"* action=*"saveemp.htm"* method= *"post"*>

<pre>

<h1><b>Employee Registration Form</b></h1></br></br>

Employee Id: <input type=*"text"* name=*"id"*/></br></br>

First Name: <input type=*"text"* name=*"fname"*/></br></br>

Last Name: <input type=*"text"* name=*"lname"*/></br></br>

<input type=*"submit"* value=*"Save Employee"*/>

</pre>

</form>

</body>

</html>

Success.jsp ;

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

</body>

</html>

Jd-servlet.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:mvc=*"http://www.springframework.org/schema/mvc"*

xsi:schemaLocation=*"http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-4.3.xsd*

*http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/context https://www.springframework.org/schema/context/spring-context-3.1.xsd"*>

<bean id=*"viewResolver"* class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

<property name=*"prefix"* value=*"/pages/"*></property>

<property name=*"suffix"* value=*".jsp"*></property>

</bean>

<context:component-scan base-package=*"com.jd.\*"*></context:component-scan>

</beans>

Controller:

**package** com.jd.controller;

**import** org.springframework.stereotype.Controller;

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

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

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

@Controller

**public** **class** EmployeeController{

@RequestMapping(value="/getempform.htm")

**public** String showEmpForm() {

System.***out***.println("Inside showEmpForm()");

**return** "empform";

}

@RequestMapping(value="/saveemp.htm", method=RequestMethod.***POST***)

**public** String saveEmp(@RequestParam("id") **int** eid, String fname, String lname) {

System.***out***.println("inside saveEmp()");

System.***out***.println(eid + fname + lname);

**return** "success";

}

}

Employee Management:

AOP :

package com.jd.aop;

import org.aspectj.lang.JoinPoint;

import org.aspectj.lang.ProceedingJoinPoint;

import org.aspectj.lang.annotation.AfterReturning;

import org.aspectj.lang.annotation.AfterThrowing;

import org.aspectj.lang.annotation.Around;

import org.aspectj.lang.annotation.Aspect;

import org.aspectj.lang.annotation.Before;

import org.aspectj.lang.annotation.Pointcut;

import org.springframework.context.annotation.EnableAspectJAutoProxy;

import org.springframework.stereotype.Component;

@Aspect

@Component

public class loggingAspect {

@Pointcut(value = "execution(\* \*.save\*(..))")

public void dummy() {

}

@Before(value = "dummy()") // instead of repeating expression we are calling method.

public void logBefore(JoinPoint jp) {

System.out.println("Inside method:"+ jp.getSignature().getName());

}

@AfterReturning(value = "dummy()", returning="r") // return only when business component executed successfully.

public void logAfter(JoinPoint jp, Object r) {

System.out.println("Exiting from the method."+ jp.getSignature().getName());

}

@AfterThrowing(value = "dummy()", throwing="e")

public void logError(Exception e) {

System.out.println("Inside exception method.");

}

/\* @Around(value = "dummy()")

public Object logAround(ProceedingJoinPoint pjp) {

System.out.println("Inside ProceedingJoinPoint method");

Object returnval = null;

try {

System.out.println("Inside ProceedingJoinPoint method : Before of Around");

returnval = pjp.proceed();

System.out.println("Inside ProceedingJoinPoint method : After of Around");

} catch (Throwable e) {

System.out.println("Inside ProceedingJoinPoint method : Afterthrowing of Around");

e.printStackTrace();

}

return returnval;

}\*/

}

Employee Controller :

package com.jd.controller;

import java.util.ArrayList;

import java.util.List;

import java.util.Map;

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

import org.springframework.stereotype.Controller;

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

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

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

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

import com.jd.model.Employee;

import com.jd.service.EmployeeService;

import com.jd.service.Iservice;

@Controller

public class EmployeeController{

@Autowired

private Iservice es;

@RequestMapping(value="/getempform.htm")

public String showEmpForm() {

System.out.println("Inside showEmpForm()");

return "empform";

}

/\* @RequestMapping(value="/saveemp.htm", method=RequestMethod.POST)

public String saveEmp(@RequestParam("id") int eid, String fname, String lname) {

System.out.println("inside saveEmp()");

System.out.println(eid + fname + lname);

return "success";

}\*/

@RequestMapping(value="/saveemp.htm", method=RequestMethod.POST)

public String saveEmp(@ModelAttribute("emp") Employee e, Map map) {

System.out.println("inside saveEmp()");

System.out.println(e.getEid() + e.getFname() + e.getLname());

map.put("emp", e);

List list= new ArrayList();

list.add(e);

es.saveEmployees(list);

return "success";

}

}

Employee Dao :

package com.jd.data;

import java.util.List;

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

import org.springframework.jdbc.core.BeanPropertyRowMapper;

import org.springframework.jdbc.core.JdbcTemplate;

import org.springframework.stereotype.Component;

import org.springframework.stereotype.Repository;

import org.springframework.transaction.PlatformTransactionManager;

import org.springframework.transaction.annotation.Propagation;

import org.springframework.transaction.annotation.Transactional;

import com.jd.model.Employee;

//@Component("dao")

@Repository("dao")

public class EmployeeDAO implements Idao {

@Autowired

private JdbcTemplate template;

private PlatformTransactionManager transactionManager;

@Override

//@Transactional(propagation = Propagation.SUPPORTS)

@Transactional(propagation = Propagation.REQUIRES\_NEW)

public void saveEmployee(Employee e) {

//System.out.println("Inside saveEmployee method of data layer.");

String sql = "insert into Employee value(?,?,?)";

template.update(sql,e.getEid(),e.getFname(),e.getLname());

}

@Override

public void modifyEmployee(Employee e) {

//System.out.println("Inside modifyEmployee method of data layer.");

String sql = "update Employee set fname=?, lname=? where Id=?";

template.update(sql,e.getFname(),e.getLname(), e.getEid());

}

@Override

public List<Employee> getAllEmployee() {

// TODO Auto-generated method stub

String sql = "select \* from Employee";

//return template.query(sql, new BeanPropertyRowMapper<Employee>(Employee.class));

// below row we have writtern if I have mismatching variable name and DB column name. And that is why we create new class "EmployeeRowMapper" to map.

return template.query(sql, new EmployeeRowMapper());

}

@Override

public Employee getEmployeebyID(int empid) {

String sql = "select \* from Employee where Id=?";

return template.queryForObject(sql, new Object[] {empid}, new EmployeeRowMapper());

}

@Override

public String getEmployeeNamebyID(int empid) {

String sql = "select fname from Employee where Id=?";

return template.queryForObject(sql, String.class, empid);

}

@Override

public int getEmployeeCount() {

String sql = "select count(\*) from Employee";

return template.queryForObject(sql, Integer.class);

}

}

Rowmapper :

package com.jd.data;

import java.sql.ResultSet;

import java.sql.SQLException;

import org.springframework.jdbc.core.RowMapper;

import com.jd.model.Employee;

// this class is for custom row mapper.

public class EmployeeRowMapper implements RowMapper<Employee> {

@Override

public Employee mapRow(ResultSet rs, int rownumber) throws SQLException {

System.out.println("Inside mapRow");

int id = rs.getInt("id");

String fn = rs.getString("fname");

String ln = rs.getString("lname");

Employee e = new Employee(id, fn, ln);

return e;

}

}

Idao :

package com.jd.data;

import java.util.List;

import com.jd.model.Employee;

public interface Idao {

void saveEmployee(Employee e);

void modifyEmployee(Employee e);

List<Employee> getAllEmployee();

Employee getEmployeebyID(int empid);

String getEmployeeNamebyID(int empid);

int getEmployeeCount();

}

Employee :

package com.jd.model;

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

import com.jd.service.EmployeeService;

public class Employee {

private int eid;

private String fname;

private String lname;

public Employee() {

super();

// TODO Auto-generated constructor stub

}

public Employee(int eid, String fname, String lname) {

super();

this.eid = eid;

this.fname = fname;

this.lname = lname;

}

public int getEid() {

return eid;

}

public void setEid(int eid) {

this.eid = eid;

}

/\*public int getId() {

return eid;

}

public void setId(int id) {

this.eid = id;

}\*/

public String getFname() {

return fname;

}

public void setFname(String fname) {

this.fname = fname;

}

public String getLname() {

return lname;

}

public void setLname(String lname) {

this.lname = lname;

}

@Override

public String toString() {

return "Employee [eid=" + eid + ", fname=" + fname + ", lname=" + lname + "]";

}

}

Service :

package com.jd.service;

import java.util.List;

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

import org.springframework.stereotype.Component;

import org.springframework.stereotype.Service;

import org.springframework.transaction.PlatformTransactionManager;

import org.springframework.transaction.annotation.Isolation;

import org.springframework.transaction.annotation.Propagation;

import org.springframework.transaction.annotation.Transactional;

import com.jd.data.Idao;

import com.jd.model.Employee;

//@Component("service")

@Service("service")

public class EmployeeService implements Iservice {

@Autowired

private Idao dao;

@Autowired

private PlatformTransactionManager transactionManager;

@Override

@Transactional(propagation = Propagation.REQUIRED,rollbackFor = Exception.class, isolation=Isolation.DEFAULT)

public void saveEmployees(List<Employee> list) {

//System.out.println("Inside saveEmployee method in service layer");

for(Employee emp : list) {

dao.saveEmployee(emp);

}

}

@Override

public void modifyEmployees(List<Employee> list) {

}

}

Iservice :

package com.jd.service;

import java.util.List;

import com.jd.model.Employee;

public interface Iservice {

void saveEmployees(List<Employee> list);

void modifyEmployees(List<Employee> list);

}

Empform.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title></title>

</head>

<body bgcolor=*"cyan"* align=*"center"*>

<form id=*"form1"* action=*"saveemp.htm"* method= *"post"*>

<pre>

<h1><b>Employee Registration Form</b></h1></br></br>

Employee Id: <input type=*"text"* name=*"eid"*/></br></br>

First Name: <input type=*"text"* name=*"fname"*/></br></br>

Last Name: <input type=*"text"* name=*"lname"*/></br></br>

<input type=*"submit"* value=*"Save Employee"*/>

</pre>

</form>

</body>

</html>

.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<h2>The following Employee is persisted</h2>

Id : ${emp.eid}

FirstName : ${emp.fname}

LastName : ${emp.lname}

</body>

</html>

Jd-servlet:

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:mvc=*"http://www.springframework.org/schema/mvc"*

xmlns:aop=*"http://www.springframework.org/schema/aop"*

xmlns:tx=*"http://www.springframework.org/schema/tx"*

xmlns:util=*"http://www.springframework.org/schema/util"*

xsi:schemaLocation=*"http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-4.3.xsd*

*http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/context https://www.springframework.org/schema/context/spring-context-3.1.xsd*

*http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-4.3.xsd*

*http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-4.3.xsd*

*http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util-4.3.xsd"*>

<bean id=*"viewResolver"* class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

<property name=*"prefix"* value=*"/pages/"*></property>

<property name=*"suffix"* value=*".jsp"*></property>

</bean>

<bean id=*"ds"* class=*"org.springframework.jdbc.datasource.DriverManagerDataSource"*>

<property name=*"driverClassName"* value=*"com.mysql.jdbc.Driver"*></property>

<property name=*"url"* value=*"jdbc:mysql://localhost:3306/emp"*></property>

<property name=*"username"* value=*"root"*></property>

<property name=*"password"* value=*"root"*></property>

</bean>

<bean id=*"template"* class=*"org.springframework.jdbc.core.JdbcTemplate"*>

<property name=*"dataSource"* ref=*"ds"*></property>

</bean>

<bean id=*"transactionManager"* class=*"org.springframework.jdbc.datasource.DataSourceTransactionManager"*>

<property name=*"dataSource"* ref=*"ds"*></property>

</bean>

<context:component-scan base-package=*"com.jd.\*"*></context:component-scan>

<tx:annotation-driven/>

<aop:aspectj-autoproxy></aop:aspectj-autoproxy>

</beans>

Web.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"* id=*"WebApp\_ID"* version=*"3.1"*>

<display-name>EmployeeManagement</display-name>

<!-- The front controller of this Spring Web application, responsible for handling all application requests -->

<servlet>

<servlet-name>jd</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<!-- Map all requests to the DispatcherServlet for handling -->

<servlet-mapping>

<servlet-name>jd</servlet-name>

<url-pattern>\*.htm</url-pattern>

</servlet-mapping>

</web-app>