\*\*\*\*\*\*\*\*\*\*\*\*Business.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**package** com.igate.aop;

**public****interface** Business {

**void** doBusiness();

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*BusinessDemo.java\*\*\*\*\*\*\*\*\*\*\***

package com.igate.aop;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class BusinessDemo {

public static void main(String[] args) {

ApplicationContext context = new ClassPathXmlApplicationContext("schemaaop.xml");

Business bc = (Business)context.getBean("myBusinessClass");

bc.doBusiness();

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Output\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Around (Before) Method Call

Before Method Call

I do what I do best, i.e sleep.

How dare you to wake me up?

Done with sleeping.

In excepiton

After Method Call

**\*\*\*\*\*\*\*\*\*\*\*\* BusinessImpl.java\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.aop;

public class BusinessImpl implements Business {

public void doBusiness() {

System.*out*.println("I do what I do best, i.e sleep.");

try {

throw new NullPointerException();

} catch (Exception e) {

System.*out*.println("How dare you to wake me up?");

}

System.*out*.println("Done with sleeping.");

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* MyAdvice.java\*\*\*\*\*\*\*\*\*\***

package com.igate.aop;

import org.aspectj.lang.ProceedingJoinPoint;

public class MyAdvice {

public void beforeMethod() {

System.*out*.println("Before Method Call");

}

public void aroundMethod(ProceedingJoinPoint joinpoint) {

System.*out*.println("Around (Before) Method Call");

try{

joinpoint.proceed();

}catch(Throwable t){}

System.*out*.println("In excepiton");

}

public void afterMethod() {

System.*out*.println("After Method Call");

}}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*log4j.properties\*\*\*\*\*\*\*\*\*\***

# For JBoss: Avoid to setup Log4J outside $JBOSS\_HOME/server/default/deploy/log4j.xml!

# For all other servers: Comment out the Log4J listener in web.xml to activate Log4J.

log4j.rootLogger=INFO, logfile

log4j.appender.stdout=org.apache.log4j.ConsoleAppender

log4j.appender.stdout.layout=org.apache.log4j.PatternLayout

log4j.appender.stdout.layout.ConversionPattern=%d %p [%c] - <%m>%n

log4j.appender.logfile=org.apache.log4j.RollingFileAppender

log4j.appender.logfile.File=basicspring.log

log4j.appender.logfile.MaxFileSize=512KB

# Keep three backup files.

log4j.appender.logfile.MaxBackupIndex=3

# Pattern to output: date priority [category] - message

log4j.appender.logfile.layout=org.apache.log4j.PatternLayout

log4j.appender.logfile.layout.ConversionPattern=%d %p [%c] - %m%n

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

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

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

*http://www.springframework.org/schema/aop*

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

<bean id=*"advice"* class=*"com.igate.aop.MyAdvice"* />

<bean id=*"myBusinessClass"* class=*"com.igate.aop.BusinessImpl"* />

<aop:config>

<aop:aspect ref=*"advice"*>

<aop:pointcut id=*"mypoint"* expression=*"execution(\* com.igate.aop.BusinessImpl.doBusiness())"*/>

<aop:around

pointcut=*"execution(\* com.igate.aop.BusinessImpl.doBusiness())"*

method=*"aroundMethod"* />

<aop:before

pointcut-ref=*"mypoint"*

method=*"beforeMethod"* />

<aop:after-returning

pointcut=*"execution(\* com.igate.aop.BusinessImpl.doBusiness())"*

method=*"afterMethod"* />

</aop:aspect>

</aop:config>

</beans>

**2)\*\*\*\*\*\*\*\*Using Annotation\*\*\*\*\*\*\*\*\*Business.java\*\*\*\*\*\*\*\*\*\*\***

package com.igate.aop;

public interface Business {

void doBusiness();

void myBusinessMethod();

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* BusinessImpl.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.aop;

import java.sql.Connection;

public class BusinessImpl implements Business{

public void doBusiness() {

System.*out*.println("I do what I do best, i.e sleep.");

try {

Thread.*sleep*(200);

} catch (InterruptedException e) {

System.*out*.println("How dare you to wake me up?");

}

System.*out*.println("Done with sleeping.");

}

public void myBusinessMethod()

{

System.*out*.println("In My BusinessMethod");

}}

**\*\*\*\*\*\*\*\*\*\*\*\*\* BusinessProfiler.java\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.aop;

import org.aspectj.lang.ProceedingJoinPoint;

import org.aspectj.lang.annotation.Around;

import org.aspectj.lang.annotation.Aspect;

import org.aspectj.lang.annotation.Before;

import org.aspectj.lang.annotation.Pointcut;

@Aspect

public class BusinessProfiler {

@Pointcut("execution(\* com.igate.aop.\*.d\*(..))")

public void businessMethods() { }

@Around("businessMethods()")

public Object profile(ProceedingJoinPoint pjp) throws Throwable {

Object obj = pjp.getTarget();

long start = System.currentTimeMillis();

System.out.println("Going to call the method.");

Object output=pjp.proceed();

System.out.println("Method execution completed.");

long elapsedTime = System.currentTimeMillis() - start;

System.out.println("Method execution time: " + elapsedTime + " milliseconds.");

return output;

}

@Pointcut("execution(\* com.igate.aop.\*.d\*(..))")

public void beforeMethod(){ }

@Before("beforeMethod()")

public void MyBeforeMethod(){

System.out.println("Applying Before advice"); } }

**\*\*\*\*\*\*\*\*\*BusinessDemo.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.aop;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class BusinessDemo {

public static void main(String[] args) {

ApplicationContext context = new ClassPathXmlApplicationContext("business.xml");

Business bc = (Business) context.getBean("myBusinessClass");

bc.doBusiness();

//bc.myBusinessMethod

}

}

**…………..Output……………………………….**

Going to call the method.

Applying Before advice

I do what I do best, i.e sleep.

Done with sleeping.

Method execution completed.

Method execution time: 203 milliseconds.

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

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

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

*http://www.springframework.org/schema/aop*

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

<!-- Enable the @AspectJ support -->

<aop:aspectj-autoproxy />

<bean id=*"businessProfiler"* class=*"com.igate.aop.BusinessProfiler"* />

<bean id=*"myBusinessClass"* class=*"com.igate.aop.BusinessImpl"* />

</beans>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*SpringMVCAOP\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Login.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**package** com.igate.demo;

**import** org.hibernate.validator.constraints.NotEmpty;

**public** **class** Login {

**private** String userName;

**private** String password;

@NotEmpty(message="username cannot be empty")

**public** String getUserName() {

**return** userName;

}

@NotEmpty(message="password cannot be empty")

**public** String getPassword() {

**return** password;

}

**public** **void** setUserName(String userName) {

**this**.userName = userName;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*LoginController.java\*\*\*\*\*\*\***

package com.igate.demo;

import javax.validation.Valid;

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

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.validation.BindingResult;

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.servlet.ModelAndView;

@Controller

public class LoginController {

@Autowired

private LoginDao loginDao;

@RequestMapping(value="/showlogin", method=RequestMethod.GET)

public String showLogin(Model model)

{

model.addAttribute("log",new Login());

return "login";

}

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

public ModelAndView validate(@ModelAttribute("log") @Valid Login log ,BindingResult result)

{

if(result.hasErrors())

return new ModelAndView("login");

System.out.println(log.getUserName());

if(loginDao.validate(log))

return new ModelAndView("success","l",log);

else

return new ModelAndView("error");

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*LoginDao.java\*\*\*\*\*\*\*\*\*\*\*\*

package com.igate.demo;

import org.apache.log4j.Logger;

import org.springframework.stereotype.Component;

@Component("loginDao")

public class LoginDao {

Logger myLog;

public LoginDao()

{

myLog = Logger.getLogger(LogInfo.class);

}

public boolean validate(Login login)

{

if(login.getUserName().equals("user") && login.getPassword().equals("user"))

{

myLog.info("User successfully Logged In");

return true;

}

else

{

myLog.info("Login FAILS!!!");

return false;

}

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*LogInfo.java\*\*\*\*\*\*\*

package com.igate.demo;

import org.apache.log4j.Logger;

import org.apache.log4j.PropertyConfigurator;

import org.apache.log4j.helpers.Loader;

import java.net.URL;

import org.aspectj.lang.annotation.Aspect;

import org.aspectj.lang.annotation.Before;

import org.aspectj.lang.annotation.Pointcut;

//@Aspect

public class LogInfo {

// @Pointcut("execution(\* com.igate.demo.\*.\*(..))")

public void configure(){ }

// @Before("configure()")

public void configureLogging() throws Throwable {

Object point = null;

Logger myLog = Logger.getLogger(LogInfo.class);

URL url=Loader.getResource("log4j.properties");

PropertyConfigurator.configure(url.getFile());

try {

System.out.println("from logging aspect: entering method");

myLog.info("Hello : It is " + new java.util.Date().toString());

System.out.println("from logging aspect: exiting method ");

} catch (Exception e) {

System.out.println("Exception occured");

}

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*src/log4j.properties\*\*\*\*\*

# For JBoss: Avoid to setup Log4J outside $JBOSS\_HOME/server/default/deploy/log4j.xml!

# For all other servers: Comment out the Log4J listener in web.xml to activate Log4J.

log4j.rootLogger=INFO, stdout

log4j.logger.org.springframework=ERROR

log4j.logger.org.hibernate=ERROR

log4j.appender.stdout=org.apache.log4j.ConsoleAppender

log4j.appender.stdout.layout=org.apache.log4j.PatternLayout

log4j.appender.stdout.layout.ConversionPattern=%d %p [%c] - <%m>%n

log4j.appender.logfile=org.apache.log4j.RollingFileAppender

log4j.appender.logfile.File=basicspring.log

log4j.appender.logfile.MaxFileSize=512KB

# Keep three backup files.

log4j.appender.logfile.MaxBackupIndex=3

# Pattern to output: date priority [category] - message

log4j.appender.logfile.layout=org.apache.log4j.PatternLayout

log4j.appender.logfile.layout.ConversionPattern=%d %p [%c] - %m%n

\*\*\*\*\*\*\*\*\*\*dispatcher-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:aop=*"http://www.springframework.org/schema/aop"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

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

*http://www.springframework.org/schema/context*

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

*http://www.springframework.org/schema/aop*

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

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

<bean id=*"logging"* class=*"com.igate.demo.LogInfo"* />

<!-- To enable Spring Components -->

<context:component-scan base-package=*"com.igate.demo"* />

<aop:config>

<aop:aspect ref=*"logging"*>

<aop:before pointcut=*"execution(\* com.igate.demo.\*.\*(..))"* method=*"configureLogging"*/>

</aop:aspect>

</aop:config>

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

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

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

</bean>

</beans>

\*\*\*\*\*\*\*\*\*\*\*web.xml\*\*\*\*\*\*\*\*\*\*

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

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://java.sun.com/xml/ns/javaee"* xmlns:web=*"http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"* xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"* id=*"WebApp\_ID"* version=*"2.5"*>

<display-name>DemoWebApp\_Login</display-name>

<welcome-file-list>

<welcome-file>index.jsp</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>dispatcher</servlet-name>

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

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

</servlet>

<servlet-mapping>

<servlet-name>dispatcher</servlet-name>

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

</servlet-mapping>

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>/WEB-INF/dispatcher-servlet.xml</param-value>

</context-param>

<listener>

<listener-class>

org.springframework.web.context.ContextLoaderListener

</listener-class>

</listener>

</web-app>

**\*\*\*\*\*\*\*\*\*\*\*\*index.jsp\*\*\*\*\*\*\*\***

<html>

<head>

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

<title>Insert title here</title>

</head>

<body>

<h1>In Index PAGE</h1>

<a href=*"showlogin.html"*>Click Here to Login</a>

</body>

</html>

**\*\*\*\*\*\*\*\*\*\*\*login.jsp\*\*\*\*\*\*\*\*\*\*\*\***

<%@ taglib uri=*"http://java.sun.com/jsp/jstl/core"* prefix=*"c"*%>

<%@ taglib uri=*"http://www.springframework.org/tags/form"* prefix=*"form"*%>

<html>

<head>

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

<title>LOGIN PAGE</title>

</head>

<body>

<h1>In Login PAGE</h1>

<c:url var=*"url"* value=*"/afterlogin.html"* />

<form:form modelAttribute=*"log"* action=*"*${url}*"* method=*"post"*>

<form:label path=*"userName"*>Username</form:label>

<form:input path=*"userName"*/>

<font color=*"red"*><form:errors path=*"userName"*/></font><br/>

<form:label path=*"password"*>Password</form:label>

<form:input path=*"password"*/>

<font color=*"red"*><form:errors path=*"password"*/></font><br/>

<input type=*"submit"* value=*"Login"*/>

</form:form>

</body>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*error.jsp\*\*\*\*\*\*\*\*\*\*\*\*\***

<html>

<head>

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

<title>Insert title here</title>

</head>

<body>

Login Credentials are WRONG!!!!!!

</body>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*success.jsp\*\*\*\*\*\*\*\*\*\*\***

<html>

<head>

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

<title>Insert title here</title>

</head>

<body>

<h1>Success PAGE</h1>

Your Info:<br/>

${l.userName}<br/>

</body>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Spring4 Testing\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Employee.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**package** com.igate.jdbc;

**public** **class** Employee {

**private** **int** eid;

**private** String enm;

**private** **double** esl;

**public** **int** getEid() {

**return** eid;

}

**public** **void** setEid(**int** eid) {

**this**.eid = eid;

}

**public** String getEnm() {

**return** enm;

}

**public** **void** setEnm(String enm) {

**this**.enm = enm;

}

**public** **double** getEsl() {

**return** esl;

}

**public** **void** setEsl(**double** esl) {

**this**.esl = esl;

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*EmployeeDao.java\*\*\*\*\*\*\*\*\*\*\***

package com.igate.jdbc;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.List;

import org.springframework.jdbc.core.JdbcTemplate;

import org.springframework.jdbc.core.RowMapper;

public class EmployeeDao {

JdbcTemplate jdbcTemplate;

public JdbcTemplate getJdbcTemplate() {

return jdbcTemplate;

}

public void setJdbcTemplate(JdbcTemplate jdbcTemplate) {

this.jdbcTemplate = jdbcTemplate;

}

public int getCount() {

int count = 0;

count = jdbcTemplate

.queryForObject("SELECT count(\*) FROM employees",Integer.class);

return count;

}

public String getEmployeeName(int eid) {

String sql = "SELECT ename FROM employees WHERE eid=?";

String name = getJdbcTemplate().queryForObject(sql, String.class,

eid);

return name;

}

public int insertRec(int eid, String enm, double esl) {

String sql = "INSERT INTO employees VALUES(?,?)";

Object[] params = new Object[] { eid, enm, esl };

return jdbcTemplate.update(sql, params);

}

public int updateRec(int eid, String enm, double esl) {

String sql = "UPDATE employees SET ename=?,esal=? WHERE eid=?";

Object[] params = new Object[] { enm, esl, eid };

int update = jdbcTemplate.update(sql, params);

return update;

}

public List getAll() {

Object[] params = new Object[] { new Double(40000.00) };

List list = jdbcTemplate.queryForList(

"SELECT \* FROM employees WHERE esal > ?", params);

return list;

}

// to retrieve the an employee given eid

public Employee getEmpByEid(int eid) {

String sql = "SELECT eid,ename,esal FROM employees WHERE eid=?";

RowMapper mapper = new RowMapper() {

public Employee mapRow(ResultSet rs, int rowNum)

throws SQLException {

Employee e = new Employee();

e.setEid(rs.getInt(1));

e.setEnm(rs.getString(2));

e.setEsl(rs.getDouble(3));

System.out.println(rowNum);

return e;

}

};

Employee employee = (Employee) jdbcTemplate.queryForObject(sql,

mapper, eid);

return employee;

}

public List<Employee> getEmployeeList() {

String sql = "SELECT \* FROM employees";

List<Employee> eList = jdbcTemplate.query(sql,

new EmployeeRowMapper());

return eList;

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*EmployeeRowMapper.java\*\*\*\*\*\*\***

package com.igate.jdbc;

import java.sql.ResultSet;

import java.sql.SQLException;

import org.springframework.jdbc.core.RowMapper;

public class EmployeeRowMapper implements RowMapper {

@Override

public Object mapRow(ResultSet rs, int line) throws SQLException {

Employee e = new Employee();

e.setEid(rs.getInt(1));

e.setEnm(rs.getString(2));

e.setEsl(rs.getDouble(3));

return e;

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*JdbcClient.java\*\*\*\*\***

package com.igate.jdbc;

import java.sql.SQLException;

import java.util.List;

import javax.sql.DataSource;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class JdbcClient {

public static void main(String[] args) throws SQLException {

// TODO Auto-generated method stub

System.setProperty("spring.profiles.active", "oracle");

ApplicationContext bf=new ClassPathXmlApplicationContext("testDao.xml");

DataSource ds=(DataSource)bf.getBean("dataSource");

//System.out.println(ds.getConnection().getSchema());

EmployeeDao eDao=(EmployeeDao)bf.getBean("employeeDao");

//execute queryForInt

int count=eDao.getCount();

System.out.println("No of Employees in the table :"+count);

String name=eDao.getEmployeeName(111);

System.out.println("Name "+name);

//execute update (insert new record)

/\*int insert=eDao.insertRec(11111,"Shiv",55555.55);

if(insert>0)

System.out.println("Insert Success");

else

System.out.println("Insert Failed");

//execute update (modify)

int update=eDao.updateRec(106, "Shiv Kumar", 77777.77);

if(update>0)

System.out.println("Update Success");

else

System.out.println("Update Failed");

\*/

//execute QueryForList

List list=eDao.getAll();

System.out.println("List of Records");

for(Object emp:list)

System.out.println(emp);

Employee employee=eDao.getEmpByEid(1105);

System.out.println("Employee Info");

System.out.println("EID :"+employee.getEid());

System.out.println("ENM :"+employee.getEnm());

System.out.println("ESL :"+employee.getEsl());

List<Employee> eList=eDao.getEmployeeList();

System.out.println("List of Employees");

for(Employee e:eList)

System.out.println(e.getEid()+" "+e.getEnm()+" "+e.getEsl());

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*MyDaoTest.java\*\*\*\*\*\*\*\*\*\***

package com.igate.testcontext;

import junit.framework.Assert;

import org.junit.Test;

import org.junit.runner.RunWith;

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

import org.springframework.test.context.ActiveProfiles;

import org.springframework.test.context.ContextConfiguration;

import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;

import com.igate.jdbc.EmployeeDao;

@RunWith(SpringJUnit4ClassRunner.class)

@ContextConfiguration("testDao.xml")

@ActiveProfiles("oracle")

public class MyDAOTest

{

@Autowired

private EmployeeDao employeeDAO;

@Test

public void testSalary()

{

System.out.println("in MyDAOTest in testSalary()" +

employeeDAO.getEmployeeName(1105));

Assert.assertNotNull(employeeDAO.getCount());

}

@Test

public void testAllEmployee()

{

System.out.println("in MyDAOTest in testAllEmployee()" );

Assert.assertNotNull(employeeDAO.getAll());

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*MyTest.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.testcontext;

import junit.framework.Assert;

import org.junit.Test;

import org.junit.runner.RunWith;

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

import org.springframework.test.context.ContextConfiguration;

import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;

import com.igate.writer.Writer;

@RunWith(SpringJUnit4ClassRunner.class)

@ContextConfiguration(locations={"testBean.xml"})

public class MyTest {

@Autowired

private Writer writer;

@Test

public void createTest() {

System.out.println("in test class");

Assert.assertNotNull(writer.getNumber());

}

@Test

public void createTest1() {

System.out.println("in test class");

Assert.assertEquals(100, writer.getNumber());

}

}

**\*\*\*\*\*\*\*\*\*\*\*com.igate.testcontext/testBean.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"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context-3.0.xsd"*>

<bean id=*"writer"* class=*"com.igate.writer.Writer"* >

<property name=*"number"* value=*"100"* />

</bean>

</beans>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*com.igate.testcontext/testDao.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=*"properties"*

class=*"org.springframework.beans.factory.config.PropertyPlaceholderConfigurer"*>

<property name=*"location"*>

<value>oracle.properties</value>

</property>

</bean>

<beans profile=*"oracle"*>

<bean id=*"dataSource"*

class=*"org.springframework.jdbc.datasource.DriverManagerDataSource"*>

<property name=*"driverClassName"* value=*"${db.driver}"* />

<property name=*"url"* value=*"${db.url}"* />

<property name=*"username"* value=*"${db.user}"* />

<property name=*"password"* value=*"${db.password}"* />

</bean>

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

<constructor-arg ref=*"dataSource"* />

</bean>

<bean id=*"employeeDao"* class=*"com.igate.jdbc.EmployeeDao"*>

<property name=*"jdbcTemplate"* ref=*"jdbcTemplate"* />

</bean>

</beans>

</beans>

**\*\*\*\*\*\*\*\*\*\*\*IWriter.java\*\*\*\*\*\*\*\***

package com.igate.writer;

public interface IWriter {

public int writer(int i);

}

**\*\*\*\*\*\*\*\*\*Writer.java\*\*\*\*\*\*\*\*\*\*\***

package com.igate.writer;

public class Writer implements IWriter{

int number;

public int writer(int i) {

System.*out*.println(i+number);

return number+i;

}

public int getNumber() {

return number;

}

public void setNumber(int number) {

this.number = number;

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*log4j.properties\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

# For JBoss: Avoid to setup Log4J outside $JBOSS\_HOME/server/default/deploy/log4j.xml!

# For all other servers: Comment out the Log4J listener in web.xml to activate Log4J.

log4j.rootLogger=INFO, stdout, logfile

log4j.appender.stdout=org.apache.log4j.ConsoleAppender

log4j.appender.stdout.layout=org.apache.log4j.PatternLayout

log4j.appender.stdout.layout.ConversionPattern=%d %p [%c] - <%m>%n

log4j.appender.logfile=org.apache.log4j.RollingFileAppender

log4j.appender.logfile.File=basicspring.log

log4j.appender.logfile.MaxFileSize=512KB

# Keep three backup files.

log4j.appender.logfile.MaxBackupIndex=3

# Pattern to output: date priority [category] - message

log4j.appender.logfile.layout=org.apache.log4j.PatternLayout

log4j.appender.logfile.layout.ConversionPattern=%d %p [%c] - %m%n

**\*\*\*\*\*\*\*\*\*\*\*\*oracle.properties\*\*\*\*\*\*\*\***

db.driver=oracle.jdbc.driver.OracleDriver

db.url=jdbc:oracle:thin:@localhost:1521:XE

db.user=system

db.password=root

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Thank You\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***