Spring injection in field using annotation(27-03-2018)

AnnotatedDemo:

Package:

Com.hexa.dao:

IDao(interface)

**package** com.hexa.dao;

**public** **interface** Idao {

**public** String ViewData();

}

jdbcDaoimpl☹java)

**package** com.hexa.dao;

**import** org.springframework.stereotype.Component;

**import** org.springframework.stereotype.Repository;

@Repository("jdbcdao")

**public** **class** JdbcDaoImpl **implements** Idao {

**public** JdbcDaoImpl() {

System.***out***.println("constructor");

}

@Override

**public** String ViewData() {

**return** "jdbc";

}

}

Hibdaoimpl(java)

**package** com.hexa.dao;

**import** org.springframework.stereotype.Repository;

@Repository("hibdao")

**public** **class** HibDaoImopl **implements** Idao {

**public** HibDaoImopl() {

// **TODO** Auto-generated constructor stub

}

@Override

**public** String ViewData() {

// **TODO** Auto-generated method stub

**return** "hibernate";

}

}

------

Com.hexa..service

Iser(interface)

**package** com.hexa.service;

**public** **interface** Iser {

String getData();

}

Serimpl(java)

**package** com.hexa.service;

**import** javax.annotation.PostConstruct;

**import** javax.annotation.PreDestroy;

**import** org.springframework.beans.factory.DisposableBean;

**import** org.springframework.beans.factory.InitializingBean;

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

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

**import** org.springframework.stereotype.Component;

**import** org.springframework.stereotype.Repository;

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

**import** com.hexa.dao.Idao;

@Service("myser")

**public** **class** SerImpl **implements** Iser,InitializingBean, DisposableBean{

**public** Idao getDao() {

**return** dao;

}

**private** Idao dao;

// setter is for Spring Ioc to do dependency Injection

@Override

**public** String getData() {

**return** dao.ViewData();

}

**public** SerImpl() {

System.***out***.println("service constructor");

}

@Autowired

@Qualifier("hibdao")

**public** **void** setDao(Idao dao) {

System.***out***.println("dao is injected");

**this**.dao = dao;

}

@Override

**public** **void** destroy() **throws** Exception {

System.***out***.println("destroy");

}

@Override

**public** **void** afterPropertiesSet() **throws** Exception {

System.***out***.println("after properties set");

}

@PostConstruct

**public** **void** Myinit() {

System.***out***.println("post construct");

}

@PreDestroy

**public** **void** Mydestroy() {

System.***out***.println("pre destroy ");

}

}

----

Com.hexa.client

clientA(java)

**package** com.hexa.client;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.annotation.AnnotationConfigApplicationContext;

**import** org.springframework.context.support.AbstractApplicationContext;

**import** com.hexa.service.Iser;

**public** **class** ClientA {

**private** **static** AbstractApplicationContext *ctx*;

**static** {

*ctx* = **new** AnnotationConfigApplicationContext(MyConfig.**class**);

}

**public** **static** **void** main(String[] args) {

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

Iser ser= *ctx*.getBean("myser",Iser.**class**);

System.***out***.println(ser.getData());

*ctx*.registerShutdownHook();

System.***out***.println("demand for second time");

*ctx*.registerShutdownHook();

}

}

Myconfig(java):

**package** com.hexa.client;

**import** org.springframework.context.annotation.ComponentScan;

**import** org.springframework.context.annotation.Configuration;

@Configuration

@ComponentScan("com.hexa")

**public** **class** MyConfig {

}

---

Pom:

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>com.hexa</groupId>

<artifactId>AnnotationDemo</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>AnnotationDemo</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<maven.compiler.source>1.8</maven.compiler.source>

<maven.compiler.target>1.8</maven.compiler.target>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>4.3.2.RELEASE</version>

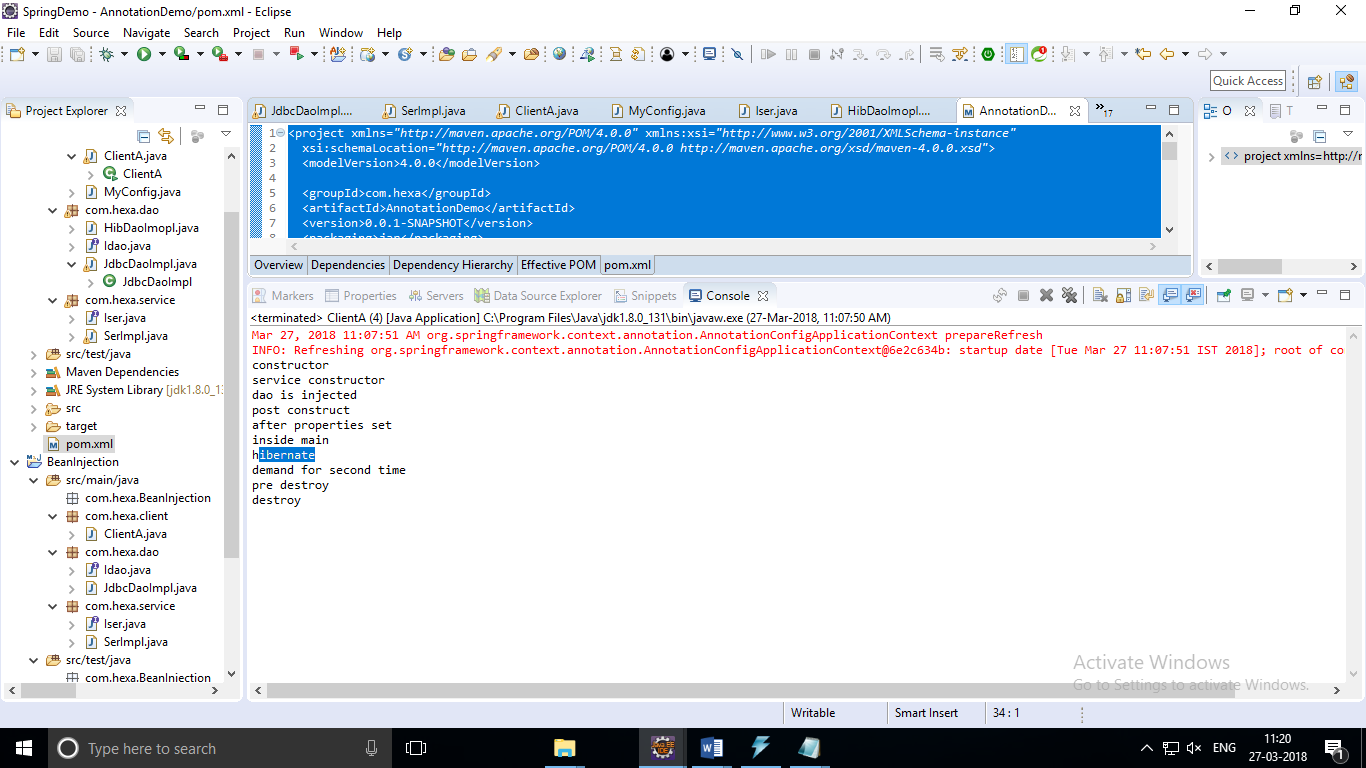
</dependency>

</dependencies>

</project>

--

Output:



SPRING JDBC:

Com.hexa.dao:

DaoImpl(java):

package com.hexa.dao;

import java.util.List;

import java.util.Map;

import org.springframework.jdbc.core.JdbcTemplate;

import org.springframework.jdbc.core.RowMapper;

import com.hexa.entity.Student;

public class DaoImpl implements IDao{

private JdbcTemplate jtemp;

private RowMapper rmap;

public void setRmap(RowMapper rmap) {

this.rmap = rmap;

}

public void setJtemp(JdbcTemplate jtemp) {

this.jtemp = jtemp;

}

@Override

public List<Map<String, Object>> getStudentsAsMap(String dept) {

String sql ="select \* from student where stu\_dept=?";

Object[] params = {dept};

List<Map<String, Object>> lst = jtemp.queryForList(sql,params);

return lst;

}

@Override

public Map<String, Object> getStudentsAsMap(int sid) {

String sql ="select \* from student where stu\_id=?";

Object[] params = {sid};

Map<String,Object> map=jtemp.queryForMap(sql,params);

return map;

}

@Override

public int updatemarks(int sid, int marks) {

String sql ="update student set stu\_marks=? where stu\_id=?";

Object[] params= {marks,sid};

int rows=jtemp.update(sql,params);

return rows;

}

@Override

public List<Student> getStudents(String dept) {

String sql ="select \* from student where stu\_dept=?";

Object[] params= {dept};

List<Student> lst =jtemp.query(sql,params,rmap);

return lst;

}

@Override

public Student getStudent(int sid) {

String sql ="select \* from student where stu\_id=?";

Object[] params= {sid};

Student lst = (Student) jtemp.queryForObject(sql,params,rmap);

return lst;

}

}

---

IDao(interface):

package com.hexa.dao;

import java.util.List;

import java.util.Map;

import com.hexa.entity.Student;

public interface IDao {

List<Map<String, Object>> getStudentsAsMap(String dept);

Map<String,Object> getStudentsAsMap(int sid);

int updatemarks(int sid,int marks);

List<Student> getStudents(String dept);

Student getStudent(int sid);

}

======

StuMapper(java)

package com.hexa.dao;

import java.sql.ResultSet;

import java.sql.SQLException;

import org.springframework.jdbc.core.RowMapper;

import com.hexa.entity.Student;

public class StuMapper implements RowMapper<Student>{

@Override

public Student mapRow(ResultSet rs, int idx) throws SQLException {

Student stu = new Student();

stu.setSid(rs.getInt("Stu\_id"));

stu.setSname(rs.getString("Stu\_name"));

stu.setMarks(rs.getInt("Stu\_marks"));

stu.setDept(rs.getString("Stu\_dept"));

stu.setDate(rs.getDate("stu\_dob"));

return stu;

}

}

===========

Com.hexa.client:

ClientA(java)

package com.hexa.client;

import java.util.List;

import java.util.Map;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.hexa.dao.IDao;

public class ClientA {

private static ApplicationContext ctx;

static {

ctx = new ClassPathXmlApplicationContext("spring-jdbc.xml");

}

public static void main(String[] args) {

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

IDao dao = ctx.getBean("mydao", IDao.class);

Map<String, Object> map1 = dao.getStudentsAsMap(2002);

System.out.println("find by student ID 1002");

System.out.println(map1.get("stu\_id"));

System.out.println(map1.get("stu\_name"));

System.out.println(map1.get("stu\_marks"));

System.out.println(map1.get("stu\_dept"));

System.out.println(map1.get("stu\_dob"));

System.out.println(map1.get("##################################################"));

System.out.println("view Student by dept");

List<Map<String, Object>> lst = dao.getStudentsAsMap("eee");

for (Map map : lst) {

System.out.println(map.get("stu\_id"));

System.out.println(map.get("stu\_name"));

System.out.println(map.get("stu\_marks"));

System.out.println(map.get("stu\_dept"));

System.out.println(map.get("stu\_dob"));

System.out.println(map.get("##################################################"));

}

}

}

====

UpdateMarks(java)

package com.hexa.client;

import java.util.Map;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.hexa.dao.IDao;

public class UpdateMarks {

private static ApplicationContext ctx;

static {

ctx = new ClassPathXmlApplicationContext("spring-jdbc.xml");

}

public static void main(String[] args) {

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

IDao dao = ctx.getBean("mydao", IDao.class);

int rows=dao.updatemarks(1002, 98);

System.out.println(rows+"no of rows updated");

Map<String, Object> map1 = dao.getStudentsAsMap(2002);

System.out.println("find by student ID 1002");

System.out.println(map1.get("stu\_id"));

System.out.println(map1.get("stu\_name"));

System.out.println(map1.get("stu\_marks"));

System.out.println(map1.get("stu\_dept"));

System.out.println(map1.get("stu\_dob"));

System.out.println(map1.get("##################################################"));

}

}

====

ViewStudent(java)

package com.hexa.client;

import java.util.List;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.hexa.dao.IDao;

import com.hexa.entity.Student;

public class ViewStudent {

private static ApplicationContext ctx;

static {

ctx = new ClassPathXmlApplicationContext("spring-jdbc.xml");

}

public static void main(String[] args) {

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

IDao dao = ctx.getBean("mydao", IDao.class);

System.out.println("=========view all students===========");

List<Student> lst = dao.getStudents("eee");

for(Student stu:lst) {

System.out.println(stu);

}

System.out.println("==========view by student id======");

System.out.println(dao.getStudent(1005));

}

}

===

Com.hexa.entity

Student(java)

**package** com.hexa.entity;

**import** java.util.Date;

**public** **class** Student {

**private** **int** sid;

**private** String sname;

**private** **int** marks;

@Override

**public** String toString() {

**return** "Student [sid=" + sid + ", sname=" + sname + ", marks=" + marks + ", dept=" + dept + ", date=" + date

+ "]";

}

**private** String dept;

**private** Date date;

**public** **int** getSid() {

**return** sid;

}

**public** **void** setSid(**int** sid) {

**this**.sid = sid;

}

**public** String getSname() {

**return** sname;

}

**public** **void** setSname(String sname) {

**this**.sname = sname;

}

**public** **int** getMarks() {

**return** marks;

}

**public** **void** setMarks(**int** marks) {

**this**.marks = marks;

}

**public** String getDept() {

**return** dept;

}

**public** **void** setDept(String dept) {

**this**.dept = dept;

}

**public** Date getDate() {

**return** date;

}

**public** **void** setDate(Date date) {

**this**.date = date;

}

}

===

Src->maim>resources->

spring-jdbc.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"*

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

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

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

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

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

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

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

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

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

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

</bean>

<bean id =*"rowmapper"* class=*"com.hexa.dao.StuMapper"*></bean>

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

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

</bean>

<bean id=*"mydao"* class=*"com.hexa.dao.DaoImpl"*>

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

<property name=*"rmap"* ref=*"rowmapper"*></property>

</bean>

</beans>

Pom.xml

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>com.hexa</groupId>

<artifactId>SpringJdbc</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>SpringJdbc</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<maven.compiler.source>1.8</maven.compiler.source>

<maven.compiler.target>1.8</maven.compiler.target>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>4.3.2.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>4.3.2.RELEASE</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

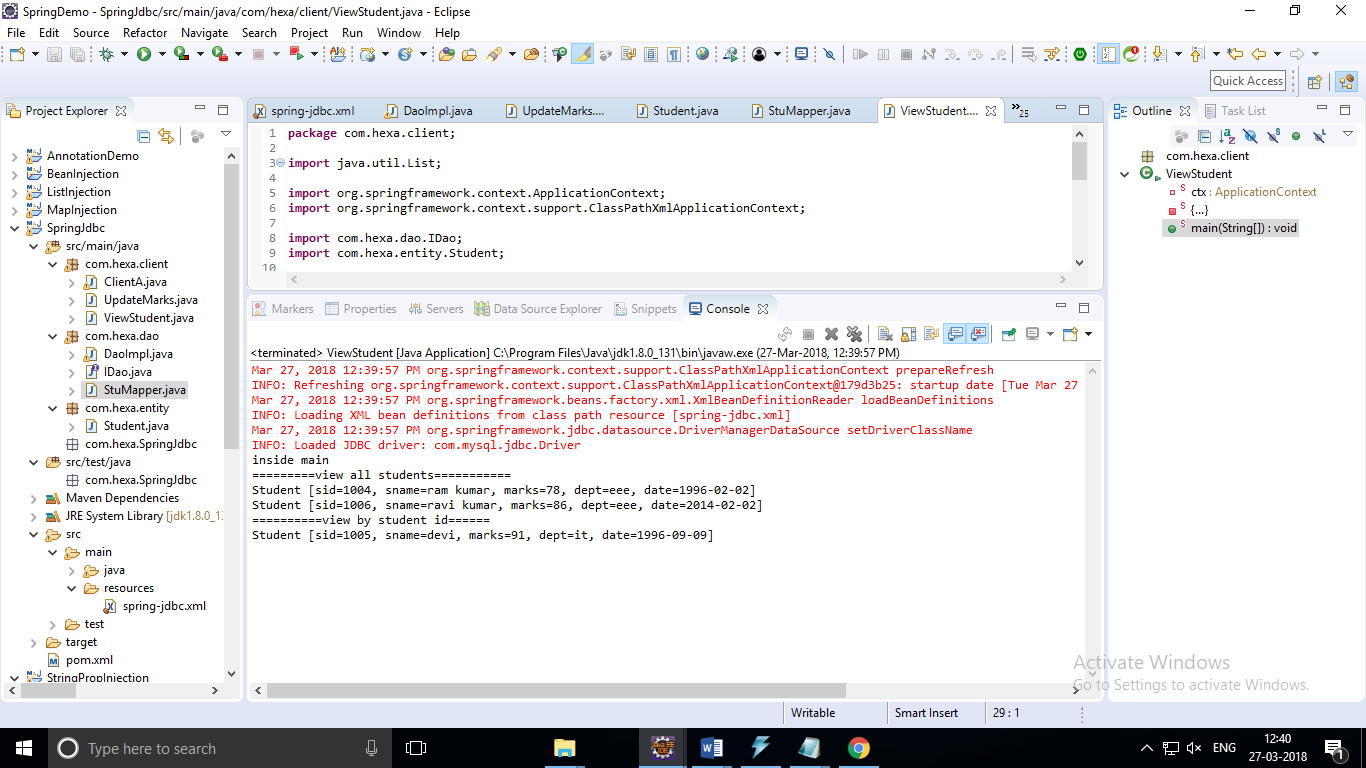
<version>5.1.6</version>

</dependency>

</dependencies>

</project>

Output:



SpringJDBCAnnotated:

com.hexa.client:

ClientA(java)

package com.hexa.client;

import java.util.List;

import java.util.Map;

import org.springframework.context.ApplicationContext;

import org.springframework.context.annotation.AnnotationConfigApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.hexa.dao.IDao;

public class ClientA {

private static ApplicationContext ctx;

static {

ctx = new AnnotationConfigApplicationContext(JdbcConfig.class);

}

public static void main(String[] args) {

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

IDao dao = ctx.getBean("mydao", IDao.class);

Map<String, Object> map1 = dao.getStudentsAsMap(2002);

System.out.println("find by student ID 1002");

System.out.println(map1.get("stu\_id"));

System.out.println(map1.get("stu\_name"));

System.out.println(map1.get("stu\_marks"));

System.out.println(map1.get("stu\_dept"));

System.out.println(map1.get("stu\_dob"));

System.out.println(map1.get("##################################################"));

System.out.println("view Student by dept");

List<Map<String, Object>> lst = dao.getStudentsAsMap("eee");

for (Map map : lst) {

System.out.println(map.get("stu\_id"));

System.out.println(map.get("stu\_name"));

System.out.println(map.get("stu\_marks"));

System.out.println(map.get("stu\_dept"));

System.out.println(map.get("stu\_dob"));

System.out.println(map.get("##################################################"));

}

}

}

==

JdbcConfig(java):

package com.hexa.client;

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

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

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.ComponentScan;

import org.springframework.context.annotation.Configuration;

import org.springframework.context.annotation.PropertySource;

import org.springframework.core.env.Environment;

import org.springframework.jdbc.core.JdbcTemplate;

import org.springframework.jdbc.datasource.DriverManagerDataSource;

@Configuration

@ComponentScan("com.hexa")

@PropertySource("classpath:hexa.properties")

public class JdbcConfig {

@Autowired

private Environment env;

@Bean(name="jdbctemp")

@Autowired

@Qualifier("mydatasource")

public JdbcTemplate getJdbcTemplate(DriverManagerDataSource ds){

JdbcTemplate jt = new JdbcTemplate();

jt.setDataSource(ds);

return jt;

}

@Bean(name="mydatasource")

public DriverManagerDataSource getDataSource(){

DriverManagerDataSource ds = new DriverManagerDataSource();

ds.setDriverClassName(env.getProperty("jdriver"));

ds.setUrl(env.getProperty("jurl"));

ds.setUsername(env.getProperty("uname"));

ds.setPassword(env.getProperty("pwd"));

return ds;

}

}

==

UpdateMarks(java):

package com.hexa.client;

import java.util.Map;

import org.springframework.context.ApplicationContext;

import org.springframework.context.annotation.AnnotationConfigApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.hexa.dao.IDao;

public class UpdateMarks {

private static ApplicationContext ctx;

static {

ctx = new AnnotationConfigApplicationContext(JdbcConfig.class);

}

public static void main(String[] args) {

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

IDao dao = ctx.getBean("mydao", IDao.class);

int rows=dao.updatemarks(1002, 98);

System.out.println(rows+"no of rows updated");

Map<String, Object> map1 = dao.getStudentsAsMap(2002);

System.out.println("find by student ID 1002");

System.out.println(map1.get("stu\_id"));

System.out.println(map1.get("stu\_name"));

System.out.println(map1.get("stu\_marks"));

System.out.println(map1.get("stu\_dept"));

System.out.println(map1.get("stu\_dob"));

System.out.println(map1.get("##################################################"));

}

}

===

ViewStudent(java):

package com.hexa.client;

import java.util.List;

import org.springframework.context.ApplicationContext;

import org.springframework.context.annotation.AnnotationConfigApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.hexa.dao.IDao;

import com.hexa.entity.Student;

public class ViewStudent {

private static ApplicationContext ctx;

static {

ctx = new AnnotationConfigApplicationContext(JdbcConfig.class);

}

public static void main(String[] args) {

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

IDao dao = ctx.getBean("mydao", IDao.class);

System.out.println("=========view all students===========");

List<Student> lst = dao.getStudents("eee");

for(Student stu:lst) {

System.out.println(stu);

}

System.out.println("==========view by student id======");

System.out.println(dao.getStudent(1005));

}

}