**Spring Annotations**

* Create a maven project and name it as ‘AnnotationDemo’
* Set build path and compiler settings
* Add spring framework in pom.xml
* Copy com.hexa.service , com.hexa.client and com.hexa.dao from beanInjection project
* Make the following changes to the class files

**package** com.hexa.dao;

**public** **interface** IDao {

**public** String viewData();

}

**package** com.hexa.dao;

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

@Component("jdbcdao")

**public** **class** JdbcDaoImpl **implements** IDao{

**public** JdbcDaoImpl() {

System.***out***.println("Jdbc Dao Constructor");

}

@Override

**public** String viewData() {

**return** "jdbc";

}

}

**package** com.hexa.service;

**public** **interface** ISer {

**public** String getData();

}

**package** com.hexa.service;

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

**import** com.hexa.dao.IDao;

@Component("myser")

**public** **class** SerImpl **implements** ISer{

**private** IDao dao;

**public** SerImpl() {

System.***out***.println("Service Constructor fires");

}

// Setter is for spring IOC to do dependency Injection

@Autowired

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

System.***out***.println("Dao is Injected");

**this**.dao = dao;

}

@Override

**public** String getData() {

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

**return** dao.viewData();

}

}

**Client Code**

**package** com.hexa.client;

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

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

@Configuration

@ComponentScan("com.hexa")

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

}

**package** com.hexa.client;

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

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

**import** com.hexa.service.ISer;

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

**private** **static** ApplicationContext *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());

}

}

**Output :**

Jdbc Dao Constructor

Service Constructor fires

Dao is Injected

Inside Main

jdbc

**Lifecycle**

Make the following chamges to the above code to understand the lifecycle

**package** com.hexa.client;

**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();

}

}

**package** com.hexa.service;

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

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

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

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

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

**import** com.hexa.dao.IDao;

//@Component("myser")

@Service("myser")

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

**private** IDao dao;

**public** SerImpl() {

System.***out***.println("Service Constructor fires");

}

// Setter is for spring IOC to do dependency Injection

@Autowired

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

System.***out***.println("Dao is Injected");

**this**.dao = dao;

}

@Override

**public** String getData() {

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

**return** dao.viewData();

}

@Override

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

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

System.***out***.println("After Destroy");

}

@Override

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

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

System.***out***.println("After Properties set");

}

}

Run the above code . the following output is obtained

Jdbc Dao Constructor

Service Constructor fires

Dao is Injected

After Properties set

Inside Main

jdbc

Mar 27, 2018 10:43:50 AM org.springframework.context.annotation.AnnotationConfigApplicationContext doClose

INFO: Closing org.springframework.context.annotation.AnnotationConfigApplicationContext@6e2c634b: startup date [Tue Mar 27 10:43:50 IST 2018]; root of context hierarchy

After Destroy

Furthermore changes

**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.stereotype.Component;

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

**import** com.hexa.dao.IDao;

//@Component("myser")

@Service("myser")

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

**private** IDao dao;

**public** SerImpl() {

System.***out***.println("Service Constructor fires");

}

// Setter is for spring IOC to do dependency Injection

@Autowired

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

System.***out***.println("Dao is Injected");

**this**.dao = dao;

}

@Override

**public** String getData() {

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

**return** dao.viewData();

}

@Override

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

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

System.***out***.println("After Destroy");

}

@PostConstruct

**public** **void** myinit() {

System.***out***.println("Post Construct");

}

@PreDestroy

**public** **void** mydestroy() {

System.***out***.println("Pre Destroy");

}

@Override

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

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

System.***out***.println("After Properties set");

}

}

**Output:**

Jdbc Dao Constructor

Service Constructor fires

Dao is Injected

Post Construct

After Properties set

Inside Main

jdbc

Demand For the second time

Mar 27, 2018 10:51:58 AM org.springframework.context.annotation.AnnotationConfigApplicationContext doClose

INFO: Closing org.springframework.context.annotation.AnnotationConfigApplicationContext@6e2c634b: startup date [Tue Mar 27 10:51:58 IST 2018]; root of context hierarchy

Pre Destroy

After Destroy

**Injection using name**

Add HibDaoImpl.class file

**package** com.hexa.dao;

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

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

@Repository("hibdao")

**public** **class** HibDaoImpl **implements** IDao{

**public** HibDaoImpl() {

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

}

@Override

**public** String viewData() {

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

**return** "Hibernate";

}

}

**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.Service;

**import** com.hexa.dao.IDao;

//@Component("myser")

@Service("myser")

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

**private** IDao dao;

**public** SerImpl() {

System.***out***.println("Service Constructor fires");

}

// Setter is for spring IOC to do dependency Injection

@Autowired

@Qualifier("jdbcdao")

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

System.***out***.println("Dao is Injected");

**this**.dao = dao;

}

@Override

**public** String getData() {

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

**return** dao.viewData();

}

@Override

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

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

System.***out***.println("After Destroy");

}

@PostConstruct

**public** **void** myinit() {

System.***out***.println("Post Construct");

}

@PreDestroy

**public** **void** mydestroy() {

System.***out***.println("Pre Destroy");

}

@Override

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

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

System.***out***.println("After Properties set");

}

}

If interface has two implementation auto iwred will get confused which implementation to choose . Hence must provide the qualifier annotation

O/P:

Hibernate constructor

Jdbc Dao Constructor

Service Constructor fires

Dao is Injected

Post Construct

After Properties set

Inside Main

jdbc

Demand For the second time

Mar 27, 2018 11:06:43 AM org.springframework.context.annotation.AnnotationConfigApplicationContext doClose

INFO: Closing org.springframework.context.annotation.AnnotationConfigApplicationContext@6e2c634b: startup date [Tue Mar 27 11:06:43 IST 2018]; root of context hierarchy

Pre Destroy

After Destroy

**SPRING JDBC**

* Create new maven proj and name it as SpringJdbc
* Add the following 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>

* Create folder resources and create a file 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=*"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"*/>

</bean>

</beans>

Create package com.hexa.dao and an interface IDao

**package** com.hexa.dao;

**import** java.util.List;

**import** java.util.Map;

**public** **interface** IDao {

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

}

**package** com.hexa.dao;

**import** java.util.List;

**import** java.util.Map;

**import** org.springframework.jdbc.core.JdbcTemplate;

**public** **class** DaoImpl **implements** IDao {

**private** JdbcTemplate jtemp;

**public** **void** setJtemp(JdbcTemplate jtemp) {

**this**.jtemp = jtemp;

}

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

String sql = "Select \* from student where dept=?";

Object[] params = {dept};

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

**return** lst;

}

}

**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**);

List<Map<String,Object>> lst = dao.getStudentAsMap("ECE");

**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("dept"));

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

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

}

}

}

O/P:

Inside Main

1001

Abhi

86

ECE

1995-10-12 00:00:00.0

-----------------

1002

Benny

86

ECE

1995-10-19 00:00:00.0

-----------------

1005

Katherine

55

ECE

1995-05-14 00:00:00.0

-----------------

1007

Ram Kumar

69

ECE

1995-06-09 00:00:00.0

-----------------

1009

Ravi Kumar

95

ECE

1995-06-04 00:00:00.0

**package** com.hexa.dao;

**import** java.util.List;

**import** java.util.Map;

**import** com.hexa.entity.Student;

**public** **interface** IDao {

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

Map<String, Object> getSTudentAsMap(**int** studentId);

**int** updateMarks(**int** studentId,**int** marks);

List<Student> getStudents(String dept);

Student getStudent(**int** studentId);

}

**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;

}

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

String sql = "Select \* from student where dept=?";

Object[] params = {dept};

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

**return** lst;

}

**public** Map<String, Object> getSTudentAsMap(**int** studentId) {

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

Object[] params = {studentId};

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

**return** map;

}

**public** **int** updateMarks(**int** studentId, **int** marks) {

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

Object[] params = {marks,studentId};

**int** updateRows = jtemp.update(sql, params);

**return** updateRows;

}

**public** List<Student> getStudents(String dept) {

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

String sql = "Select \* from student where dept=?";

Object[] params = {dept};

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

**return** lst;

}

**public** Student getStudent(**int** studentId) {

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

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

Object[] params = {studentId};

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

**return** stu;

}

}

**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>{

**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("dept"));

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

**return** stu;

}

}

**package** com.hexa.entity;

**import** java.util.Date;

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

**private** **int** sId;

**private** String sName;

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

**private** String dept;

**private** Date dob;

**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 getDob() {

**return** dob;

}

**public** **void** setDob(Date dob) {

**this**.dob = dob;

}

@Override

**public** String toString() {

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

**return** sId + " " + sName + " " + marks + " " + dept + " " + dob;

}

}

**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**);

System.***out***.println("List All Students By Department");

List<Map<String, Object>> lst = dao.getStudentAsMap("ECE");

**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("dept"));

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

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

}

System.***out***.println("Find by student Id ");

Map<String, Object> map = dao.getSTudentAsMap(1002);

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("dept"));

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

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

}

}

**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** 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**);

System.***out***.println("Update Marks");

**int** updRes = dao.updateMarks(1002, 95);

**if**(updRes > 0) {

System.***out***.println("Rows Updated Succesfully");

Map<String, Object> map = dao.getSTudentAsMap(1002);

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("dept"));

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

} **else** {

System.***out***.println("Updation error");

}

}

}

**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;

**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 Students By Dept");

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

**for**(Student stu : lst) {

System.***out***.println(stu);

}

System.***out***.println("View Students by student id");

Student stu = dao.getStudent(1002);

System.***out***.println(stu);

}

}

<?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 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"*/>

</bean>

</beans>

O/P:

JDBAC Annotations

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);

System.out.println("List All Students By Department");

List<Map<String, Object>> lst = dao.getStudentAsMap("ECE");

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("dept"));

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

System.out.println("-----------------");

}

System.out.println("Find by student Id ");

Map<String, Object> map = dao.getSTudentAsMap(1002);

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("dept"));

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

System.out.println("-----------------");

}

}

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;

}

}

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 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);

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

int updRes = dao.updateMarks(1002, 95);

if(updRes > 0) {

System.out.println("Rows Updated Succesfully");

Map<String, Object> map = dao.getSTudentAsMap(1002);

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("dept"));

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

} else {

System.out.println("Updation error");

}

}

}

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;

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 Students By Dept");

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

for(Student stu : lst) {

System.out.println(stu);

}

System.out.println("View Students by student id");

Student stu = dao.getStudent(1002);

System.out.println(stu);

}

}

package com.hexa.dao;

import java.util.List;

import java.util.Map;

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

import org.springframework.jdbc.core.JdbcTemplate;

import org.springframework.jdbc.core.RowMapper;

import org.springframework.stereotype.Repository;

import com.hexa.entity.Student;

@Repository("mydao")

public class DaoImpl implements IDao {

@Autowired

private JdbcTemplate jtemp;

@Autowired

private RowMapper rmap;

public void setRmap(RowMapper rmap) {

this.rmap = rmap;

}

public void setJtemp(JdbcTemplate jtemp) {

this.jtemp = jtemp;

}

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

String sql = "Select \* from student where dept=?";

Object[] params = {dept};

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

return lst;

}

public Map<String, Object> getSTudentAsMap(int studentId) {

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

Object[] params = {studentId};

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

return map;

}

public int updateMarks(int studentId, int marks) {

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

Object[] params = {marks,studentId};

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

return updateRows;

}

public List<Student> getStudents(String dept) {

// TODO Auto-generated method stub

String sql = "Select \* from student where dept=?";

Object[] params = {dept};

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

return lst;

}

public Student getStudent(int studentId) {

// TODO Auto-generated method stub

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

Object[] params = {studentId};

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

return stu;

}

}

package com.hexa.dao;

import java.util.List;

import java.util.Map;

import com.hexa.entity.Student;

public interface IDao {

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

Map<String, Object> getSTudentAsMap(int studentId);

int updateMarks(int studentId,int marks);

List<Student> getStudents(String dept);

Student getStudent(int studentId);

}

package com.hexa.dao;

import java.sql.ResultSet;

import java.sql.SQLException;

import org.springframework.jdbc.core.RowMapper;

import org.springframework.stereotype.Component;

import com.hexa.entity.Student;

@Component("stumapper")

public class StuMapper implements RowMapper<Student>{

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("dept"));

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

return stu;

}

}

jdriver=com.mysql.jdbc.Driver

jurl=jdbc:mysql://localhost:3306/test

uname=root

pwd=root

Spring integration with hibernate

<?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=*"sessFactory"*

class=*"org.springframework.orm.hibernate5.LocalSessionFactoryBean"*>

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

<property name=*"hibernateProperties"*>

<props>

<prop key=*"hibernate.show\_sql"*>true</prop>

<prop key=*"hibernate.dialect"*>org.hibernate.dialect.MySQLDialect</prop>

<prop key=*"hibernate.format\_sql"*>true</prop>

</props>

</property>

<property name=*"packagesToScan"*>

<list>

<value>com.hexa.entity</value>

</list>

</property>

</bean>

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

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

</bean>

</beans>

Spring Annotation

package com.hexa.client;

import java.util.Properties;

import javax.sql.DataSource;

import org.hibernate.SessionFactory;

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

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.datasource.DriverManagerDataSource;

import org.springframework.orm.hibernate5.HibernateTransactionManager;

import org.springframework.orm.hibernate5.LocalSessionFactoryBean;

import org.springframework.transaction.annotation.EnableTransactionManagement;

@Configuration

//@EnableTransactionManagement

@PropertySource(value="classpath:hexa.properties")

@ComponentScan("com.hexa")

public class HibernateConfig {

@Autowired

private Environment env;

@Bean

public DriverManagerDataSource getdataSource(){

DriverManagerDataSource ds = new DriverManagerDataSource();

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

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

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

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

return ds;

}

@Bean

@Autowired

public LocalSessionFactoryBean getSessionFactory(DriverManagerDataSource ds){

System.out.println(ds.getUsername());

LocalSessionFactoryBean fac = new LocalSessionFactoryBean();

fac.setDataSource(ds);

fac.setHibernateProperties(getProperties());

fac.setPackagesToScan(new String[]{"com.hexa.entity"});

return fac;

}

public Properties getProperties(){

Properties props = new Properties();

props.setProperty("hibernate.dialect", env.getProperty("dialect"));

props.setProperty("hibernate.show\_sql", env.getProperty("showsql"));

props.setProperty("hibernate.format\_sql", env.getProperty("formatsql"));

return props;

}

/\*@Bean

@Autowired

public HibernateTransactionManager getTxmanger(SessionFactory sfac){

HibernateTransactionManager tx = new HibernateTransactionManager();

tx.setSessionFactory(sfac);

return tx;

}\*/

}

package com.hexa.dao;

import java.util.List;

import com.hexa.entity.Dept;

import com.hexa.entity.Emp;

public interface EmpDao {

List<Dept> getDepartments();

List<Emp> getEmployees();

List<Emp> getEmployees(int did);

Emp getEmployee(int eid);

int addEmployee(Emp emp);

}

**package** com.hexa.dao;

**import** java.util.List;

**import** org.hibernate.HibernateException;

**import** org.hibernate.Query;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

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

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

**import** com.hexa.entity.Dept;

**import** com.hexa.entity.Emp;

@Repository("mydao")

**public** **class** EmpDaoImpl **implements** EmpDao {

**private** SessionFactory sfac;

@Autowired

**public** **void** setSfac(SessionFactory sfac) {

**this**.sfac = sfac;

}

@Override

**public** List<Dept> getDepartments() {

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

Session sess = sfac.openSession();

Query qry = sess.createQuery("from Dept d");

List<Dept> lst = qry.list();

sess.close();

**return** lst;

}

@Override

**public** List<Emp> getEmployees() {

Session sess = sfac.openSession();

Query qry = sess.createQuery("from Emp e.dept");

List<Emp> lst = qry.list();

sess.close();

**return** lst;

}

@Override

**public** List<Emp> getEmployees(**int** did) {

Session sess = sfac.openSession();

Query qry = sess.createQuery("from Emp e fetch e.dept d where d.deptId = ?");

qry.setInteger(0, did);

List<Emp> lst = qry.list();

sess.close();

**return** lst;

}

@Override

**public** Emp getEmployee(**int** eid) {

Session sess = sfac.openSession();

Query qry = sess.createQuery("from Emp e fetch e.dept d where e.empId = ?");

qry.setInteger(0, eid);

Emp emp = (Emp) qry.uniqueResult();

sess.close();

**return** emp;

}

@Override

**public** **int** addEmployee(Emp emp) {

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

Session sess = **null**;

Transaction tx = **null**;

**try** {

sess = sfac.openSession();

tx = sess.beginTransaction();

sess.save(emp);

tx.commit();

**return** 1;

} **catch**(HibernateException e) {

System.***out***.println(e);

tx.rollback();

} **finally** {

sess.close();

}

**return** 0;

}

}

package com.hexa.entity;

import java.util.Set;

import javax.persistence.CascadeType;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.FetchType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.OneToMany;

import javax.persistence.Table;

@Entity

@Table(name="hexa\_dept")

public class Dept {

@Id

@Column(name="did")

private int deptId;

@Column(name="dname", length=20)

private String deptName;

@OneToMany(mappedBy = "dept")

private Set<Emp> elist;

public Set<Emp> getElist() {

return elist;

}

public void setElist(Set<Emp> elist) {

this.elist = elist;

}

public int getDeptId() {

return deptId;

}

public void setDeptId(int deptId) {

this.deptId = deptId;

}

@Override

public String toString() {

// TODO Auto-generated method stub

return deptId + " " + deptName;

}

public String getDeptName() {

return deptName;

}

public void setDeptName(String deptName) {

this.deptName = deptName;

}

}

package com.hexa.entity;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.ManyToOne;

import javax.persistence.Table;

@Entity

@Table(name="hexa\_employee")

public class Emp {

@Override

public String toString() {

// TODO Auto-generated method stub

return empId + " " + empName + " " + sal;

}

@Id

@Column(name="emp\_id")

private int empId;

@Column(name="emp\_name")

private String empName;

@Column(name="emp\_pwd")

private String pwd;

@Column(name="emp\_img")

private String img;

@Column(name="emp\_sal")

private double sal;

@ManyToOne

@JoinColumn(name = "dept\_id", referencedColumnName = "did")

private Dept dept;

public Dept getDept() {

return dept;

}

public void setDept(Dept dept) {

this.dept = dept;

}

public int getEmpId() {

return empId;

}

public void setEmpId(int empId) {

this.empId = empId;

}

public String getEmpName() {

return empName;

}

public void setEmpName(String empName) {

this.empName = empName;

}

public String getPwd() {

return pwd;

}

public void setPwd(String pwd) {

this.pwd = pwd;

}

public String getImg() {

return img;

}

public void setImg(String img) {

this.img = img;

}

public double getSal() {

return sal;

}

public void setSal(double sal) {

this.sal = sal;

}

}

<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>SpringHibernateAnnotation</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>SpringHibernateAnnotation</name>

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

<properties>

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

</properties>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.0.1.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>5.0.1.RELEASE</version>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.1.0.Final</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

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

<version>5.1.6</version>

</dependency>

</dependencies>

</project>

driver=com.mysql.jdbc.Driver

url=jdbc:mysql://localhost:3306/test

uname=root

pwd=root

showsql=true

formatsql=true

dialect=org.hibernate.dialect.MySQLDialect

hbm2ddl=auto

SPRING JPA

Create a maven project

Add 5 dependencies

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.0.1.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>5.0.1.RELEASE</version>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.1.0.Final</version>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-entitymanager</artifactId>

<version>5.1.0.Final</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

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

<version>5.1.6</version>

</dependency>

Create a package com.hexa.entity and create a student class

package com.hexa.entity;

import java.util.Date;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name = "student")

public class Student {

@Id

@Column(name = "stu\_id")

private int stuId;

@Column(name = "stu\_name", length = 45)

private String stuName;

@Column(name = "dept", length = 20)

private String dept;

@Column(name = "stu\_dob")

private Date dob;

@Column(name = "stu\_marks")

private int marks;

public int getStuId() {

return stuId;

}

public void setStuId(int stuId) {

this.stuId = stuId;

}

public String getStuName() {

return stuName;

}

public void setStuName(String stuName) {

this.stuName = stuName;

}

public String getDept() {

return dept;

}

public void setDept(String dept) {

this.dept = dept;

}

public Date getDob() {

return dob;

}

public void setDob(Date dob) {

this.dob = dob;

}

public int getMarks() {

return marks;

}

public void setMarks(int marks) {

this.marks = marks;

}

@Override

public String toString() {

return stuId + " " + stuName + " " + marks + " " + dept + " " + dob;

}

}

Create com.hexa.dao

**package** com.hexa.dao;

**import** java.util.List;

**import** javax.persistence.EntityManager;

**import** javax.persistence.PersistenceContext;

**import** javax.persistence.Query;

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

**import** org.springframework.transaction.annotation.Transactional;

**import** com.hexa.entity.Student;

@Repository("mydao")

**public** **class** DaoImpl **implements** IDao{

@PersistenceContext

**private** EntityManager em;

@Override

**public** Student getStudent(**int** studentId) {

Student stu = em.find(Student.**class**,studentId);

**return** stu;

}

@Override

**public** List<Student> getStudents(String dept) {

Query qry = em.createQuery("from Student s where s.dept = ?");

qry.setParameter(1,dept); //Parameter index starts from 1

List<Student> lst = qry.getResultList();

**return** lst;

}

@Override

@Transactional(readOnly = **false**)

**public** **int** addStudent(Student student) {

em.persist(student);

**return** 1;

}

@Override

@Transactional(readOnly = **false**)

**public** **int** removeStudent(**int** studentId) {

Student stu = em.find(Student.**class**, studentId);

em.remove(stu);

**return** 1;

}

@Override

@Transactional(readOnly = **false**)

**public** **int** updateStudent(Student student) {

em.merge(student);

**return** 1;

}

}

**package** com.hexa.dao;

**import** java.util.List;

**import** com.hexa.entity.Student;

**public** **interface** IDao {

Student getStudent(**int** studentId);

List<Student> getStudents(String dept);

**int** addStudent(Student student);

**int** removeStudent(**int** studentId);

**int** updateStudent(Student student);

}

Create com.hexa.client

package com.hexa.client;

import java.util.Properties;

import javax.persistence.EntityManagerFactory;

import org.hibernate.SessionFactory;

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

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.datasource.DriverManagerDataSource;

import org.springframework.orm.hibernate5.HibernateTransactionManager;

import org.springframework.orm.hibernate5.LocalSessionFactoryBean;

import org.springframework.orm.jpa.JpaTransactionManager;

import org.springframework.orm.jpa.JpaVendorAdapter;

import org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean;

import org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter;

import org.springframework.transaction.PlatformTransactionManager;

import org.springframework.transaction.annotation.EnableTransactionManagement;

@Configuration

@PropertySource(value = "classpath:hexa.properties")

@EnableTransactionManagement

@ComponentScan("com.hexa")

public class JPAConfig {

@Autowired

private Environment env;

@Bean

public LocalContainerEntityManagerFactoryBean getEntityMangerFactory(){

LocalContainerEntityManagerFactoryBean lcemf = new LocalContainerEntityManagerFactoryBean();

lcemf.setDataSource(getDataSource());

lcemf.setPackagesToScan(new String[]{"com.hexa.entity"});

lcemf.setJpaVendorAdapter(jpaVendorAdapter());

lcemf.setJpaProperties(jpaProperties());

return lcemf;

}

@Bean

public JpaVendorAdapter jpaVendorAdapter() {

HibernateJpaVendorAdapter hibernateJpaVendorAdapter = new HibernateJpaVendorAdapter();

return hibernateJpaVendorAdapter;

}

@Bean(name = "dataSource")

public DriverManagerDataSource getDataSource() {

DriverManagerDataSource ds = new DriverManagerDataSource();

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

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

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

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

return ds;

}

public Properties jpaProperties() {

Properties properties = new Properties();

properties.put("hibernate.dialect", env.getRequiredProperty("dialect"));

properties.put("hibernate.show\_sql", env.getRequiredProperty("showsql"));

properties.put("hibernate.format\_sql", env.getRequiredProperty("formatsql"));

return properties;

}

@Bean

@Autowired

public JpaTransactionManager transactionManager(EntityManagerFactory emf) {

JpaTransactionManager txManager = new JpaTransactionManager();

txManager.setEntityManagerFactory(emf);

return txManager;

}

}

**package** com.hexa.client;

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

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

**import** com.hexa.dao.IDao;

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

**private** **static** ApplicationContext *ctx*;

**static** {

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

}

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

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

System.***out***.println(dao.getStudent(1001));

}

}

OutPut:

Hibernate:

select

student0\_.stu\_id as stu\_id1\_0\_0\_,

student0\_.dept as dept2\_0\_0\_,

student0\_.stu\_dob as stu\_dob3\_0\_0\_,

student0\_.stu\_marks as stu\_mark4\_0\_0\_,

student0\_.stu\_name as stu\_name5\_0\_0\_

from

student student0\_

where

student0\_.stu\_id=?

1001 Abhi 86 ECE 1995-10-12 00:00:00.0