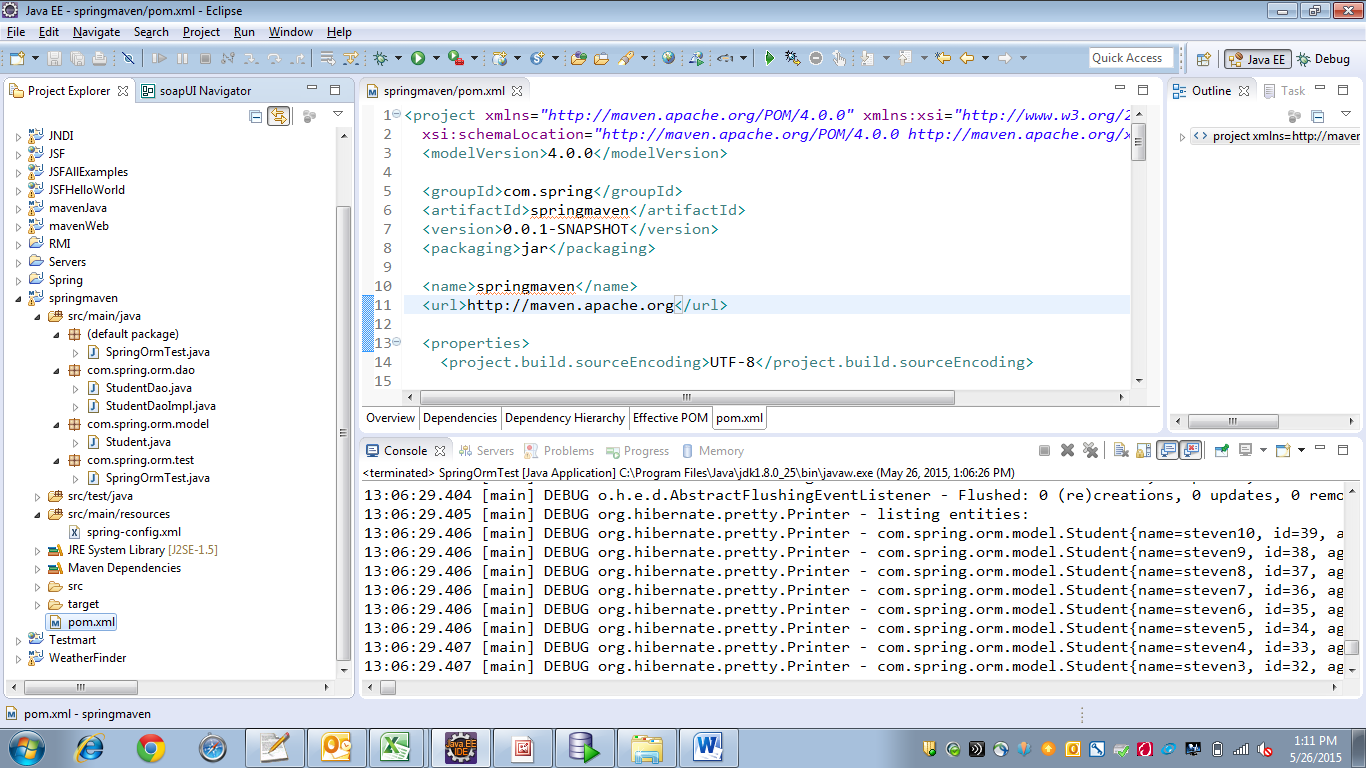
**Spring-ORM (Hibernate) integrations**

Steps to be followed:

1. Write our hibernate entity with annotations
2. Create a data source org.apache.commons.dbcp.BasicDataSources. Inject driverClassName, url, username, password to this class
3. Inject data source, mapping class name, hibernate properties to org.springframework.orm.hibernate3.annotation.AnnotationSessionFactory class
4. Inject AnnotationSessionFactory to *org.springframework.orm.hibernate3.HibernateTemplate* class.
5. Declare hibernateTemplate property in our dao class. Inject HibernateTemplate to our dao class.
6. In HibenateTemplate we have following method to perform db operations:

|  |  |  |
| --- | --- | --- |
| **No.** | **Method** | **Description** |
| 1) | void persist(Object entity) | persists the given object. |
| 2) | Serializable save(Object entity) | persists the given object and returns id. |
| 3) | void saveOrUpdate(Object entity) | persists or updates the given object. If id is found, it updates the record otherwise saves the record. |
| 4) | void update(Object entity) | updates the given object. |
| 5) | void delete(Object entity) | deletes the given object on the basis of id. |
| 6) | Object get(Class entityClass, Serializable id) | returns the persistent object on the basis of given id. |
| 7) | Object load(Class entityClass, Serializable id) | returns the persistent object on the basis of given id. |
| 8) | List loadAll(Class entityClass) | returns the all the persistent objects. |

**Create Maven project: see the project structure below:**



**pom.xml 🡪 for spring 4 and hibernate 3 dependencies**

<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.spring</groupId>

<artifactId>springmaven</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>springmaven</name>

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

<properties>

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

<!-- Spring -->

<spring-framework.version>4.0.3.RELEASE</spring-framework.version>

<!-- Hibernate / JPA -->

<!-- <hibernate.version>4.3.5.Final</hibernate.version> -->

<hibernate.version>3.6.9.Final</hibernate.version>

<!-- Logging -->

<logback.version>1.0.13</logback.version>

<slf4j.version>1.7.5</slf4j.version>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<!-- Spring and Transactions -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${spring-framework.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-tx</artifactId>

<version>${spring-framework.version}</version>

</dependency>

<!-- Spring ORM support -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>${spring-framework.version}</version>

</dependency>

<!-- Logging with SLF4J & LogBack -->

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>${slf4j.version}</version>

<scope>compile</scope>

</dependency>

<dependency>

<groupId>ch.qos.logback</groupId>

<artifactId>logback-classic</artifactId>

<version>${logback.version}</version>

<scope>runtime</scope>

</dependency>

<!-- Hibernate -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-entitymanager</artifactId>

<version>${hibernate.version}</version>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>${hibernate.version}</version>

</dependency>

<!-- Oracle jdbc driver -->

<dependency>

<groupId>ojdbc14</groupId>

<artifactId>ojdbc14</artifactId>

<version>1.0</version>

<scope>system</scope>

<systemPath>C:\jar files\ojdbc14.jar</systemPath>

</dependency>

<dependency>

<groupId>commons-dbcp</groupId>

<artifactId>commons-dbcp</artifactId>

<version>1.2.2</version>

</dependency>

</dependencies>

</project>

**Student.java 🡪 Persistable entity**

**package** com.spring.orm.model;

**import** java.io.Serializable;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.Table;

@SuppressWarnings("serial")

@Entity

@Table(name="STUDENT", schema="TEST")

**public** **class** Student **implements** Serializable{

**private** Integer id;

**private** String name;

**private** Integer age;

@Id

@GeneratedValue(strategy=GenerationType.***SEQUENCE***)

@Column(name="ID")

**public** Integer getId() {

**return** id;

}

**public** **void** setId(Integer id) {

**this**.id = id;

}

@Column(name="NAME", length=50, updatable=**true**, nullable=**true**, unique=**false**)

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

@Column(name="AGE", length=3, updatable=**true**, nullable=**true**, unique=**false**)

**public** Integer getAge() {

**return** age;

}

**public** **void** setAge(Integer age) {

**this**.age = age;

}

@Override

**public** String toString() {

**return** "[name: "+**this**.name+", age: "+**this**.age+"]";

}

}

**Spring-config.xml 🡪 for DI**

<?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"*

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*

*http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-4.0.xsd"*>

<bean id=*"datasource"* class=*"org.apache.commons.dbcp.BasicDataSource"*>

<property name=*"driverClassName"* value=*"oracle.jdbc.driver.OracleDriver"*></property>

<property name=*"url"* value=*"jdbc:oracle:thin:@localhost:1521:icp"*></property>

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

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

</bean>

<bean id=*"hibernateSessionFactory"* class=*"org.springframework.orm.hibernate3.annotation.AnnotationSessionFactoryBean"* scope=*"singleton"*>

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

<property name=*"annotatedClasses"*>

<list>

<value>com.spring.orm.model.Student</value>

</list>

</property>

<property name=*"hibernateProperties"*>

<props>

<prop key=*"hibernate.dialect"*>org.hibernate.dialect.Oracle10gDialect</prop>

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

<prop key=*"hibernate.hbm2ddl.auto"*>update</prop>

</props>

</property>

</bean>

<bean id=*"template"* class=*"org.springframework.orm.hibernate3.HibernateTemplate"*>

<property name=*"sessionFactory"* ref=*"hibernateSessionFactory"*></property>

</bean>

<bean id=*"dao"* class=*"com.spring.orm.dao.StudentDaoImpl"*>

<property name=*"sessionFactory"* ref=*"hibernateSessionFactory"*></property>

<property name=*"hibernateTemplate"* ref=*"template"*></property>

</bean>

</beans>

**StudentDao.java 🡪 Dao interface**

**package** com.spring.orm.dao;

**import** java.util.List;

**import** com.spring.orm.model.Student;

**public** **interface** StudentDao {

**public** **abstract** **void** insertStudent(Integer id, String name, Integer age);

**public** **abstract** **void** insertStudents();

**public** **abstract** **void** updateStudent(Integer id, String updatedName);

**public** **abstract** **void** deleteStudent(Integer id);

**public** **abstract** List<Student> list();

}

**StudentDaoImpl.java 🡪 Dao impl class**

package com.spring.orm.dao;

import java.util.List;

import org.hibernate.SessionFactory;

import org.springframework.orm.hibernate3.HibernateTemplate;

import com.spring.orm.model.Student;

public class StudentDaoImpl implements StudentDao {

private SessionFactory sessionFactory;

private HibernateTemplate hibernateTemplate;

public HibernateTemplate getHibernateTemplate() {

return hibernateTemplate;

}

public void setHibernateTemplate(HibernateTemplate hibernateTemplate) {

this.hibernateTemplate = hibernateTemplate;

}

public SessionFactory getSessionFactory() {

return sessionFactory;

}

public void setSessionFactory(SessionFactory sessionFactory) {

this.sessionFactory = sessionFactory;

}

@Override

public void insertStudent(Integer id, String name, Integer age) {

Student student = new Student();

student.setId(id);

student.setName(name);

student.setAge(age);

getHibernateTemplate().save(student);

}

@Override

public void insertStudents() {

Student student = null;

for(int i=1;i<=10;i++){

student = new Student();

student.setName("steven"+i);

student.setAge(i\*5);

getHibernateTemplate().save(student);

}

}

@Override

public void updateStudent(Integer id, String updatedName) {

Student student = getHibernateTemplate().get(Student.class, id);

student.setName(updatedName);

getHibernateTemplate().update(student);

}

@Override

public void deleteStudent(Integer id) {

Student student = new Student();

student.setId(id);

getHibernateTemplate().delete(student);

}

@Override

public List<Student> list() {

List<Student> list = getHibernateTemplate().loadAll(Student.class);

return list;

}

}

**Test class:**

**package** com.spring.orm.test;

**import** java.util.List;

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

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

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

**import** com.spring.orm.dao.StudentDao;

**import** com.spring.orm.model.Student;

**public** **class** SpringOrmTest {

**private** AbstractApplicationContext context = (AbstractApplicationContext) getApplicationContext();

**private** StudentDao dao = (StudentDao) context.getBean("dao");

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

SpringOrmTest test = **new** SpringOrmTest();

test.listStudents();

test.closeApplicationContext(test.context);

}

**public** **void** insertStudent(){

dao.insertStudent(1, "steven", 21);

}

**public** **void** insertStudents(){

dao.insertStudents();

}

**public** **void** updateStudent(Integer id, String updatedName){

dao.updateStudent(id, updatedName);

}

**public** **void** deleteStudent(Integer id){

dao.deleteStudent(id);

}

**public** **void** listStudents(){

List<Student> students = dao.list();

**for**(Student student: students)

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

}

**private** ApplicationContext getApplicationContext(){

**return** **new** ClassPathXmlApplicationContext("spring-config.xml");

}

**private** **void** closeApplicationContext(AbstractApplicationContext context){

**if**(context!=**null**)

context.close();

}

}