

File→new→project→maven→select archetype quickstart →finish

1. Add Dependencies

2. Spring core 5.2.3
3. Spring context 5.2.3
4. Spring ORM 5.2.3
5. Hibernate core 5.5.9
6. Mysql connector 5.1.49

```
<!-- https://mvnrepository.com/artifact/org.springframework/spring-core -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-core</artifactId>
    <version>5.2.3.RELEASE</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-context -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>5.2.3.RELEASE</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-orm -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-orm</artifactId>
    <version>5.2.3.RELEASE</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->
<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-core</artifactId>
    <version>5.5.9.Final</version>
</dependency>

<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
<dependency>
```

```
<groupId>mysql</groupId>  
<artifactId>mysql-connector-java</artifactId>  
<version>5.1.49</version>  
</dependency>
```

2. Adding Configuration File

Src/main/java → rightclick → new → xml → config.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:tx="http://www.springframework.org/schema/tx"
       xmlns:mvc="http://www.springframework.org/schema/mvc"
       xsi:schemaLocation="http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc.xsd
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-4.0.xsd">

</beans>
```

3. Create two packages under main package net.spring.com

1. net.spring.com.dao --- to keep dao classes
2. net.spring.com.entities ----to keep all entity class files

4. Create a entity under 'net.spring.com.entities' and add getter/setters, constructors

```
package net.spring.com.entities;
```

```
import javax.persistence.Entity;
import javax.persistence.Table;
```

```
@Entity
@Table (name = "Student_Details")
public class Student
{
    private int StudentID;
    private String StudentName;
    private String StudentCity;

    public Student(int studentID, String studentName, String studentCity) {
        super();
    }
}
```

```

        StudentID = studentID;
        StudentName = studentName;
        StudentCity = studentCity;
    }

    public Student() {

    }

    public int getStudentID() {
        return StudentID;
    }

    public void setStudentID(int studentID) {
        StudentID = studentID;
    }

    public String getStudentName() {
        return StudentName;
    }

    public void setStudentName(String studentName) {
        StudentName = studentName;
    }

    public String getStudentCity() {
        return StudentCity;
    }

    public void setStudentCity(String studentCity) {
        StudentCity = studentCity;
    }

}

```

5. Create StudentDao class under 'net.spring.com.dao'

```

package net.spring.com.dao;
import org.springframework.orm.hibernate5.HibernateTemplate;
import net.spring.com.entities.Student;

public class StudentDao
{
    public HibernateTemplate hibernateTemplate;

    public int insert(Student student)
    {

```

```
Integer i=(Integer)this.hibernateTemplate.save(student);  
return i;  
}  
}
```

6. Configuration of Spring

```
<?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:tx="http://www.springframework.org/schema/tx"
  xmlns:mvc="http://www.springframework.org/schema/mvc"
  xsi:schemaLocation="http://www.springframework.org/schema/mvc
    http://www.springframework.org/schema/mvc/spring-mvc.xsd
    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-4.0.xsd">

  <tx:annotation-driven/>

  <bean class="org.springframework.jdbc.datasource.DriverManagerDataSource" name="ds">
    <property name="driverClassName" value="com.mysql.jdbc.Driver"></property>
    <property name="url" value="jdbc:mysql://localhost:3306/nilesh"></property>
    <property name="username" value="root"></property>
    <property name="password" value="root"></property>
  </bean>

  <bean class="org.springframework.orm.hibernate5.LocalSessionFactoryBean" name="factory">
    <property name="dataSource" ref="ds"></property>

    <property name="hibernateProperties">
      <props>
        <prop key="hibernate.Dialect">org.hibernate.dialect.MySQL5Dialect</prop>
        <prop key="hibernate.show_sql">true</prop>
        <prop key="hibernate.hbm2ddl.auto">update</prop>
      </props>
    </property>

    <property name="annotatedClasses">
      <list>
        <value>net.spring.com.entities.Student</value>
      </list>
    </property>
  </bean>

  <bean class="org.springframework.orm.hibernate5.HibernateTemplate" name="hibernateTemplate">
    <property name="sessionFactory" ref="factory"></property>
  </bean>

  <bean class="net.spring.com.dao.StudentDao" name="studentDao">
    <property name="hibernateTemplate" ref="hibernateTemplate"></property>
  </bean>

  <bean class="org.springframework.orm.hibernate5.HibernateTransactionManager"
    name="transactionManager">
    <property name="sessionFactory" ref="factory"></property>
  </bean>

</beans>
```

The diagram illustrates the dependencies between the beans defined in the XML configuration:

- The **dataSource** bean (of type `org.springframework.jdbc.datasource.DriverManagerDataSource`) is referenced by the **factory** bean.
- The **factory** bean (of type `org.springframework.orm.hibernate5.LocalSessionFactoryBean`) is referenced by both the **hibernateTemplate** and **transactionManager** beans.
- The **hibernateTemplate** bean (of type `org.springframework.orm.hibernate5.HibernateTemplate`) is referenced by the **studentDao** bean.

7. App.java

```
package net.spring.com;

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

import net.spring.com.dao.StudentDao;
import net.spring.com.entities.Student;

/**
 * Hello world!
 */
public class App
{
    public static void main( String[] args )
    {
        System.out.println( "Hello World!" );

        ApplicationContext context= new ClassPathXmlApplicationContext("config.xml");
        StudentDao studentDao= context.getBean("studentDao",StudentDao.class);

        Student st=new Student(123,"Nilesh","Satara");

        int i= studentDao.insert(st);

        System.out.println("Record inserted.." +i);
    }
}
```

8. Run the app.java

9. Error

If you face error =

Exception in thread "main" [org.springframework.dao.InvalidDataAccessApiUsageException](#): Write operations are not allowed in read-only mode (FlushMode.MANUAL): Turn your Session into FlushMode.COMMIT/AUTO or remove 'readOnly' marker from transaction definition.

Solution:

1. Add `<tx:annotation-driven/>` in config.xml file
2. Add following code in config.xml file under `<beans>` tag

```
<bean class="org.springframework.orm.hibernate5.HibernateTransactionManager"
name="transactionManager">
    <property name="sessionFactory" ref="factory"></property>
</bean>
```

3. Give annotation `@Transactional` to your insert method

Menu Driven - App.java

```
package net.spring.com;

import java.io.BufferedInputStream;
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.Iterator;
import java.util.List;

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

import net.spring.com.dao.StudentDao;
import net.spring.com.entities.Student;

/**
 * Hello world!
 *
 */
public class App
{
    public static void main( String[] args )
    {
        System.out.println( "Hello World!" );

        ApplicationContext context= new ClassPathXmlApplicationContext("config.xml");
        StudentDao studentDao= context.getBean("studentDao",StudentDao.class);

        //Student st=new Student(123,"Nilesh","Satara");
        //int i= studentDao.insert(st);
        //System.out.println("Record inserted.." +i);

        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        boolean go=true;
        while (go)
        {
            System.out.println("Press 1- Insert");
            System.out.println("Press 2- Select All");
            System.out.println("Press 3- Select Single Student");
            System.out.println("Press 4- Delete");
            System.out.println("Press 5- Update");
            System.out.println("Press 6- Break/Exit");

            try {

                Integer ch=Integer.parseInt(br.readLine()) ;

                switch (ch) {
                    case 1:

                        System.out.println("Enter Student ID");
                        int uid=Integer.parseInt(br.readLine());

                        System.out.println("Enter Student Name");
                        String uname=br.readLine();

                        System.out.println("Enter Student City");
                        String ucity=br.readLine();

                        Student s =new Student(uid,uname,ucity);

                        int r=studentDao.insert(s);

                        System.out.println("Report Inserted");
                        break;

                    case 2:
                        System.out.println("***** Student Details *****");
```

```

        List<Student> getallstudent = studentDao.getallstudent();

        for(Student st:getallstudent)
        {
            System.out.println("Student Id:"+st.getStudentID());
            System.out.println("Student Name: "+st.getStudentName());
            System.out.println("Student City: "+st.getStudentCity());
            System.out.println("_____");
        }
        System.out.println("*****");
        break;

    case 3:
        System.out.println("***** Single Student Data*****");

        System.out.println("Enter Student ID");
        int userid=Integer.parseInt(br.readLine());

        Student getstudent = studentDao.getstudent(userid);

        System.out.println("Student Id:"+getstudent.getStudentID());
        System.out.println("Student Name: "+getstudent.getStudentName());
        System.out.println("Student City: "+getstudent.getStudentCity());
        System.out.println("_____");
        System.out.println("*****");
        break;

    case 4:

        System.out.println("Enter Student ID");
        int userdeleteid=Integer.parseInt(br.readLine());
        studentDao.deletestudent(userdeleteid);
        System.out.println("Record Deleted");
        break;

    case 5:
        //update logic

        break;

    default:
        go=false;
        break;
    }

    } catch (Exception e) {
        System.out.println("Enter Correct Input");
    }

}

}

```

Menu Driven – StudentDao.java

```
package net.spring.com.dao;
import java.util.List;
import javax.transaction.Transactional;
import org.springframework.orm.hibernate5.HibernateTemplate;
import net.spring.com.entities.Student;

public class StudentDao
{
    public HibernateTemplate hibernateTemplate;

    @Transactional
    public int insert(Student student)
    {
        Integer i=(Integer)this.hibernateTemplate.save(student);
        return i;
    }

    //get a student
    public Student getstudent(int StudentID)
    {
        Student student= this.hibernateTemplate.get(Student.class, StudentID);
        return student;
    }

    //get all record
    public List<Student> getallstudent()
    {
        List<Student> students=this.hibernateTemplate.loadAll(Student.class);
        return students;
    }

    //delete a record
    @Transactional
    public void deletestudent(int StudentID)
    {
        Student DeStud= this.hibernateTemplate.get(Student.class,StudentID);
        this.hibernateTemplate.delete(DeStud);
    }

    //update
    @Transactional
    public void updatestudent(Student studID)
    {
        this.hibernateTemplate.update(studID);
    }

    public HibernateTemplate getHibernateTemplate() {
        return hibernateTemplate;
    }
    public void setHibernateTemplate(HibernateTemplate hibernateTemplate) {
        this.hibernateTemplate = hibernateTemplate;
    }
}
```