1. Create a maven project and add following contents in pom.xml <properties>

<hibernate.version>5.4.21.Final</hibernate.version>

<hibernate.validator>5.4.1.Final</hibernate.validator>

</properties>

<dependencies>

<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core --> <dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

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

</dependency>

<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-validator --> <dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-validator</artifactId>

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

</dependency>

<!-- https://mvnrepository.com/artifact/org.slf4j/slf4j-api --> <dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-log4j12</artifactId>

<version>1.7.29</version>

</dependency>

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

<groupId>mysql</groupId>

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

<version>8.0.11</version>

</dependency>

</dependencies>

2. Download and add all jar files

Right click on project > run as > maven build > install

3. Add a file hibernate.cfg.xml in src/main/resources folder

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

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD//EN"

"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd"> <!-- Version 8 MySQL hiberante-cfg.xml example for Hibernate 5 --> <hibernate-configuration>

<session-factory>

<property name="connection.driver\_class">com.mysql.cj.jdbc.Driver</property> <!-- property name="connection.driver\_class">com.mysql.jdbc.Driver</property --> <property name="connection.url">jdbc:mysql://localhost:3306/test?useSSL=false</property> <property name="connection.username">root</property>

<property name="connection.password">root123</property>

<property name="dialect">org.hibernate.dialect.MySQL8Dialect</property> <property name="connection.pool\_size">1</property>

<!--property name="dialect">org.hibernate.dialect.MySQLDialect</property--> <!-- <property name="current\_session\_context\_class">thread</property> --> <property name="show\_sql">true</property>

<property name="format\_sql">true</property>

<property name="hbm2ddl.auto">update</property>

<!-- mapping class" / -->

<!-- <mapping class="com.demo.beans.Department"/>

<mapping class="com.demo.beans.Product"/>

<mapping class="com.demo.beans.User"/> -->

<!-- <mapping resource="product.hbm.xml"/> -->

<mapping resource="myuser.hbm.xml"/>

</session-factory>

</hibernate-configuration>

4. Inside beans package create MyUser and add uid,name,email

public class MyUser {

private int uid;

private String uname;

private String email;

public MyUser() {

super();

}

public MyUser(int uid, String uname, String email) {

super();

this.uid = uid;

this.uname = uname;

this.email = email;

}

//add getter and setter and toString method

}

5. Create a MyUser.hbm.xml file and store it in folder sr/main/resources write following code in it.

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

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD//EN"

"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping>

<class name="com.demo.beans.MyUser" table="myuser11">

<id name="uid">

<generator class="assigned"></generator>

</id>

<property name="uname" type="string" column="name" not-null="true"></property> <property name="email" type="string"></property>

</class>

</hibernate-mapping>

6. Add <mapping resource> entry in hibernate.cfg.xml file

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import org.hibernate.cfg.Configuration;

import com.demo.beans.MyUser;

public class TestUserObject {

public static void main(String[] args) {

//object is in transient state

//MyUser u1=new MyUser(13,"Rajan","aaaa@jskdf");

//step 1 create session factory object

SessionFactory sf=new Configuration().configure().buildSessionFactory(); //step2 create session

Session sess=sf.openSession();

//sf.getCurrentSession();

Transaction tr=sess.beginTransaction();

sess.save(u1);

tr.commit();

sess.close();

Session sess1=sf.openSession();

//sf.getCurrentSession();

Transaction tr1=sess1.beginTransaction();

MyUser u2=sess1.get(MyUser.class,13);

tr1.commit();

System.out.println(u2);

sess1.close();

sf.close();

}

}