Address DTO , DAO and DAOImpl

1.Inside the model folder of the backendproject , add the class , Address.java.

package net.sports.ZenSportsBackEnd.model;

import java.io.Serializable;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.validation.constraints.Size;

@Entity

public class Address implements Serializable {

public static long getSerialversionuid() {

return serialVersionUID;

}

/\*\*

\*

\*/

private static final long serialVersionUID = 5470478102236229185L;

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int addressId;

@Size(min=1, max=30,message="address1 should be between 1 to 30 characters long")

private String addAddress1;

@Size(min=1, max=30,message="address2 should be between 1 to 30 characters long")

private String addAddress2;

@Size(min=1, max=30,message="city should be between 1 to 30 characters long")

private String addCity;

@Size(min=1, max=30,message="state should be between 1 to 30 characters long")

private String addState;

@Size(min=1, max=6,message="zip code should be between 1 to 6 characters long")

private String addZipCode;

@Size(min=1, max=30,message="country should be between 1 to 30 characters long")

private String addCountry;

private boolean shipping;

private int userId;

public int getUserId() {

return userId;

}

public void setUserId(int userId) {

this.userId = userId;

}

public Address() {

super();

// TODO Auto-generated constructor stub

}

public Address(int addressId, String addAddress1, String addAddress2, String addCity, String addState,

String addZipCode, String addCountry) {

super();

this.addressId = addressId;

this.addAddress1 = addAddress1;

this.addAddress2 = addAddress2;

this.addCity = addCity;

this.addState = addState;

this.addZipCode = addZipCode;

this.addCountry = addCountry;

this.shipping = false;

}

public int getAddressId() {

return addressId;

}

public void setAddressId(int addressId) {

this.addressId = addressId;

}

public String getAddAddress1() {

return addAddress1;

}

public void setAddAddress1(String addAddress1) {

this.addAddress1 = addAddress1;

}

public String getAddAddress2() {

return addAddress2;

}

public void setAddAddress2(String addAddress2) {

this.addAddress2 = addAddress2;

}

public String getAddCity() {

return addCity;

}

public void setAddCity(String addCity) {

this.addCity = addCity;

}

public String getAddState() {

return addState;

}

public void setAddState(String addState) {

this.addState = addState;

}

public String getAddZipCode() {

return addZipCode;

}

public void setAddZipCode(String addZipCode) {

this.addZipCode = addZipCode;

}

public String getAddCountry() {

return addCountry;

}

public void setAddCountry(String addCountry) {

this.addCountry = addCountry;

}

public boolean isShipping() {

return shipping;

}

public void setShipping(boolean shipping) {

this.shipping = shipping;

}

}

2.Inside the dao folder of backendproject , add the interface , IAddressDAO.

package net.sports.ZenSportsBackEnd.dao;

import java.util.List;

import net.sports.ZenSportsBackEnd.model.Address;

public interface IAddressDAO {

public List<Address> getAllAddress();

public Address getAddress(int id);

public boolean deleteAddress(int id);

public boolean addAddress(Address a);

public boolean updateAddress(Address a);

}

3.Inside the daoimpl folder of the backend project , add a class , AddressDAOImpl.

package net.sports.ZenSportsBackEnd.daoimpl;

import java.util.List;

import org.hibernate.SessionFactory;

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

import org.springframework.stereotype.Repository;

import org.springframework.transaction.annotation.Transactional;

import net.sports.ZenSportsBackEnd.dao.IAddressDAO;

import net.sports.ZenSportsBackEnd.model.Address;

@Repository("addressDAO")

@Transactional

public class AddressDAOImpl implements IAddressDAO

{

@Autowired

private SessionFactory sessionFactory;

@Override

public Address getAddress(int id) {

return sessionFactory.getCurrentSession().get(Address.class, Integer.valueOf(id));

}

@Override

public List<Address> getAllAddress() {

return sessionFactory.getCurrentSession().createQuery("FROM Address", Address.class).getResultList();

}

@Override

public boolean deleteAddress(int id) {

try {

sessionFactory.getCurrentSession().delete(getAddress(id));

return true;

} catch (Exception ex) {

ex.printStackTrace();

return false;

}

}

@Override

public boolean addAddress(Address a) {

try {

sessionFactory.getCurrentSession().persist(a);

return true;

} catch (Exception ex) {

ex.printStackTrace();

return false;

}

}

@Override

public boolean updateAddress(Address a) {

try {

sessionFactory.getCurrentSession().update(a);

return true;

} catch (Exception ex) {

ex.printStackTrace();

return false;

}

}

}

4.Inside the test folder of the backendproject , add a java class ,AddressTestCase.

package net.sports.ZenSportsBackEnd.test;

import static org.junit.Assert.\*;

import org.junit.BeforeClass;

import org.junit.Test;

import org.springframework.context.annotation.AnnotationConfigApplicationContext;

import net.sports.ZenSportsBackEnd.dao.IAddressDAO;

import net.sports.ZenSportsBackEnd.model.Address;

public class AddressTestCase {

private static AnnotationConfigApplicationContext context;

private static IAddressDAO addressDAO;

private Address address;

@BeforeClass

public static void init() {

context = new AnnotationConfigApplicationContext();

context.scan("net.sports.ZenSportsBackEnd");

context.refresh();

addressDAO=(IAddressDAO)context.getBean("addressDAO");

}

@Test

public void testAddAddress()

{

address=new Address();

address.setAddAddress1("98-Y");

address.setAddAddress2("Vasant Kunj");

address.setAddCity("Delhi");

address.setAddCountry("India");

address.setAddState("Delhi");

address.setAddZipCode("110010");

address.setUserId(1);

assertEquals("Successfully added a address inside the table .",true,addressDAO.addAddress(address));

}

// @Test

// public void testUpdateAddress()

// {

// address=new Address();

// address.setAddAddress1("98-Y");

// address.setAddAddress2("Vasant Kunj");

// address.setAddCity("New Delhi");

// address.setAddCountry("India");

// address.setAddState("Delhi");

// address.setAddZipCode("110036");

// address.setUserId(1);

//

// assertEquals("Successfully added a address inside the table .",true,addressDAO.updateAddress(address));

// }

//

// @Test

// public void testGetAddress()

// {

// address=addressDAO.getAddress(1);

// assertEquals("Retrieving data based on id from table","Vasant Kunj",address.getAddAddress2());

// }

//

// @Test

// public void testDeleteAddress()

// {

// assertEquals("Deleting address based on id from table",true,addressDAO.deleteAddress(1));

// }

}