**BookShop\src\java\com\sharanamvaishali\dao**

**AuthorDAO.txt**

package com.sharanamvaishali.dao;

import com.sharanamvaishali.model.Author;

import com.sharanamvaishali.model.Book;

import java.util.List;

public interface AuthorDAO {

public void saveAuthor(Author author);

public List<Author> listAuthor();

public void removeAuthor(Integer authorNo);

public Author getAuthorById(Integer authorNo);

public List<Author> listOurAuthors();

public List<Book> getAllBooksByAuthor(Integer authorNo);

}

AuthorDAOImpl.txt

package com.sharanamvaishali.dao;

import com.sharanamvaishali.model.Author;

import com.sharanamvaishali.model.Book;

import java.util.List;

import org.hibernate.SessionFactory;

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

import org.springframework.stereotype.Repository;

@Repository

public class AuthorDAOImpl implements AuthorDAO {

@Autowired

private SessionFactory sessionFactory;

@Override

public void saveAuthor(Author author) {

sessionFactory.getCurrentSession().merge(author);

}

@Override

public List<Author> listAuthor() {

return sessionFactory.getCurrentSession().createQuery("from Author").list();

}

@Override

public void removeAuthor(Integer authorNo) {

Author author = (Author) sessionFactory.getCurrentSession().load(Author.class, authorNo);

if (null != author) {

sessionFactory.getCurrentSession().delete(author);

}

}

@Override

public Author getAuthorById(Integer authorNo) {

return (Author) sessionFactory.getCurrentSession().get(Author.class, authorNo);

}

@Override

public List<Author> listOurAuthors() {

return sessionFactory.getCurrentSession().createQuery("from Author").setMaxResults(10).list();

}

@Override

public List<Book> getAllBooksByAuthor(Integer authorNo) {

return sessionFactory.getCurrentSession().createQuery("FROM Book WHERE author1No = " + authorNo + " OR author2No = " + authorNo + " OR author3No = " + authorNo + " OR author4No = " + authorNo).list();

}

}

BookDAO.txt

package com.sharanamvaishali.dao;

import com.sharanamvaishali.model.Book;

import java.util.List;

public interface BookDAO {

public void saveBook(Book book);

public List<Book> listBook();

public void removeBook(Integer bookNo);

public Book getBookById(Integer bookNo);

public List<Book> listNewReleases();

public List<Book> listUpdatedBooks();

public List<Book> listTopTitles();

public void updateHits(Integer bookNo);

}

BookDAOImpl.txt

package com.sharanamvaishali.dao;

import com.sharanamvaishali.model.Book;

import com.sharanamvaishali.service.BookService;

import java.util.List;

import org.hibernate.SessionFactory;

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

import org.springframework.stereotype.Repository;

@Repository

public class BookDAOImpl implements BookDAO {

@Autowired

private SessionFactory sessionFactory;

@Autowired

private BookService bookService;

@Override

public void saveBook(Book book) {

sessionFactory.getCurrentSession().merge(book);

}

@Override

public List<Book> listBook() {

return sessionFactory.getCurrentSession().createQuery("from Book").list();

}

@Override

public void removeBook(Integer bookNo) {

Book book = (Book) sessionFactory.getCurrentSession().load(Book.class, bookNo);

if (null != book) {

sessionFactory.getCurrentSession().delete(book);

}

}

@Override

public Book getBookById(Integer bookNo) {

return (Book) sessionFactory.getCurrentSession().get(Book.class, bookNo);

}

@Override

public List<Book> listNewReleases() {

return sessionFactory.getCurrentSession().createQuery("from Book ORDER BY year DESC").setMaxResults(10).list();

}

@Override

public List<Book> listUpdatedBooks() {

return sessionFactory.getCurrentSession().createQuery("from Book WHERE edition <> 'First' ORDER BY year DESC").setMaxResults(10).list();

}

@Override

public List<Book> listTopTitles() {

return sessionFactory.getCurrentSession().createQuery("from Book ORDER BY hits DESC").setMaxResults(10).list();

}

@Override

public void updateHits(Integer bookNo) {

Book book = bookService.getBookById(bookNo);

book.setHits(book.getHits()+1);

sessionFactory.getCurrentSession().merge(book);

}

}

CategoryDAO.txt

package com.sharanamvaishali.dao;

import com.sharanamvaishali.model.Book;

import com.sharanamvaishali.model.Category;

import java.util.List;

public interface CategoryDAO {

public void saveCategory(Category category);

public List<Category> listCategory();

public void removeCategory(Integer categoryNo);

public Category getCategoryById(Integer categoryNo);

public List<Book> getAllBooksByCategory(Integer categoryNo);

}

CategoryDAOImpl.txt

package com.sharanamvaishali.dao;

import com.sharanamvaishali.model.Book;

import com.sharanamvaishali.model.Category;

import java.util.List;

import org.hibernate.SessionFactory;

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

import org.springframework.stereotype.Repository;

@Repository

public class CategoryDAOImpl implements CategoryDAO {

@Autowired

private SessionFactory sessionFactory;

@Override

public void saveCategory(Category category) {

sessionFactory.getCurrentSession().merge(category);

}

@Override

public List<Category> listCategory() {

return sessionFactory.getCurrentSession().createQuery("from Category ORDER BY category").list();

}

@Override

public void removeCategory(Integer categoryNo) {

Category category = (Category) sessionFactory.getCurrentSession().load(Category.class, categoryNo);

if (null != category) {

sessionFactory.getCurrentSession().delete(category);

}

}

@Override

public Category getCategoryById(Integer categoryNo) {

return (Category) sessionFactory.getCurrentSession().get(Category.class, categoryNo);

}

@Override

public List<Book> getAllBooksByCategory(Integer categoryNo) {

return sessionFactory.getCurrentSession().createQuery("FROM Book WHERE categoryNo = " + categoryNo).list();

}

}

CommonDAO.txt

package com.sharanamvaishali.dao;

import java.sql.Blob;

public interface CommonDAO {

public Blob getBlob(byte[] is);

}

CommonDAOImpl.txt

package com.sharanamvaishali.dao;

import java.sql.Blob;

import org.hibernate.SessionFactory;

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

import org.springframework.stereotype.Repository;

@Repository

public class CommonDAOImpl implements CommonDAO {

@Autowired

private SessionFactory sessionFactory;

@Override

public Blob getBlob(byte[] is) {

Blob blob = sessionFactory.getCurrentSession().getLobHelper().createBlob(is);

return blob;

}

}

**CountryDAO.txt**

package com.sharanamvaishali.dao;

import com.sharanamvaishali.model.Country;

import java.util.List;

public interface CountryDAO {

public void saveCountry(Country country);

public List<Country> listCountry();

public void removeCountry(Integer countryNo);

public Country getCountryById(Integer countryNo);

}

**CountryDAOImpl.txt**

package com.sharanamvaishali.dao;

import com.sharanamvaishali.model.Country;

import java.util.List;

import org.hibernate.SessionFactory;

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

import org.springframework.stereotype.Repository;

@Repository

public class CountryDAOImpl implements CountryDAO {

@Autowired

private SessionFactory sessionFactory;

@Override

public void saveCountry(Country country) {

sessionFactory.getCurrentSession().merge(country);

}

@Override

public List<Country> listCountry() {

return sessionFactory.getCurrentSession().createQuery("from Country").list();

}

@Override

public void removeCountry(Integer countryNo) {

Country country = (Country) sessionFactory.getCurrentSession().load(Country.class, countryNo);

if (null != country) {

sessionFactory.getCurrentSession().delete(country);

}

}

@Override

public Country getCountryById(Integer countryNo) {

return (Country) sessionFactory.getCurrentSession().get(Country.class, countryNo);

}

}

**PublisherDAO.txt**

package com.sharanamvaishali.dao;

import com.sharanamvaishali.model.Book;

import com.sharanamvaishali.model.Publisher;

import java.util.List;

public interface PublisherDAO {

public void savePublisher(Publisher publisher);

public List<Publisher> listPublisher();

public void removePublisher(Integer publisherNo);

public Publisher getPublisherById(Integer publisherNo);

public List<Publisher> listOurPublishers();

public List<Book> getAllBooksByPublisher(Integer publisherNo);

}

**PublisherDAOImpl.txt**

package com.sharanamvaishali.dao;

import com.sharanamvaishali.model.Book;

import com.sharanamvaishali.model.Publisher;

import java.util.List;

import org.hibernate.SessionFactory;

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

import org.springframework.stereotype.Repository;

@Repository

public class PublisherDAOImpl implements PublisherDAO {

@Autowired

private SessionFactory sessionFactory;

@Override

public void savePublisher(Publisher publisher) {

sessionFactory.getCurrentSession().merge(publisher);

}

@Override

public List<Publisher> listPublisher() {

return sessionFactory.getCurrentSession().createQuery("from Publisher").list();

}

@Override

public void removePublisher(Integer publisherNo) {

Publisher publisher = (Publisher) sessionFactory.getCurrentSession().load(Publisher.class, publisherNo);

if (null != publisher) {

sessionFactory.getCurrentSession().delete(publisher);

}

}

@Override

public Publisher getPublisherById(Integer publisherNo) {

return (Publisher) sessionFactory.getCurrentSession().get(Publisher.class, publisherNo);

}

@Override

public List<Publisher> listOurPublishers() {

return sessionFactory.getCurrentSession().createQuery("from Publisher").setMaxResults(10).list();

}

@Override

public List<Book> getAllBooksByPublisher(Integer publisherNo) {

return sessionFactory.getCurrentSession().createQuery("FROM Book WHERE publisherNo = " + publisherNo).list();

}

}

**SearchDAO.txt**

package com.sharanamvaishali.dao;

import com.sharanamvaishali.model.PopularSearch;

import java.util.List;

public interface SearchDAO {

public List searchResults(String searchCriteria);

public List searchAllResults();

public List listPopularSearches();

public void savePopularSearch(PopularSearch popularSeacrh);

public int getTotalPopularSearches();

public void deletePopularSearches();

}

**SearchDAOImpl.txt**

package com.sharanamvaishali.dao;

import com.sharanamvaishali.model.PopularSearch;

import java.util.Iterator;

import java.util.List;

import org.hibernate.SessionFactory;

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

import org.springframework.stereotype.Repository;

@Repository

public class SearchDAOImpl implements SearchDAO {

@Autowired

private SessionFactory sessionFactory;

@Override

public List searchResults(String searchCriteria) {

return sessionFactory.getCurrentSession().createSQLQuery("SELECT DISTINCT BookNo, BookName, Year, Synopsis"

+ " FROM Books, Authors, Publishers "

+ "WHERE Authors.AuthorNo = Books.Author1No "

+ "OR Authors.AuthorNo = Books.Author2No "

+ "OR Authors.AuthorNo = Books.Author3No "

+ "OR Authors.AuthorNo = Books.Author4No "

+ "OR Publishers.PublisherNo = Books.PublisherNo "

+ "AND (BookName LIKE '%"+ searchCriteria + "%' "

+ "OR ISBN LIKE '%"+ searchCriteria + "%' "

+ "OR Edition LIKE '%"+ searchCriteria + "%' "

+ "OR Year LIKE '%"+ searchCriteria + "%' "

+ "OR Synopsis LIKE '%"+ searchCriteria + "%' "

+ "OR AboutAuthors LIKE '%"+ searchCriteria + "%' "

+ "OR TopicsCovered LIKE '%"+ searchCriteria + "%' "

+ "OR ContentsCDROM LIKE '%"+ searchCriteria + "%' "

+ "OR Cost LIKE '%"+ searchCriteria + "%' "

+ "OR FirstName LIKE '%"+ searchCriteria + "%' "

+ "OR LastName LIKE '%"+ searchCriteria + "%' "

+ "OR PublisherName LIKE '%"+ searchCriteria + "%')")

.list();

}

@Override

public List searchAllResults() {

return sessionFactory.getCurrentSession().createSQLQuery("SELECT DISTINCT BookNo, BookName, Year, Synopsis"

+ " FROM Books, Authors, Publishers "

+ "WHERE Authors.AuthorNo = Books.Author1No "

+ "OR Authors.AuthorNo = Books.Author2No "

+ "OR Authors.AuthorNo = Books.Author3No "

+ "OR Authors.AuthorNo = Books.Author4No "

+ "OR Publishers.PublisherNo = Books.PublisherNo")

.list();

}

@Override

public List listPopularSearches() {

return sessionFactory.getCurrentSession().createSQLQuery("SELECT Value, COUNT(\*) AS Weight FROM PopularSearches GROUP BY Value").list();

}

@Override

public void savePopularSearch(PopularSearch popularSearch) {

sessionFactory.getCurrentSession().save(popularSearch);

}

@Override

public int getTotalPopularSearches() {

Object TransactionNo = sessionFactory.getCurrentSession().createSQLQuery("SELECT COUNT(\*) AS Total FROM PopularSearches").addScalar("Total").uniqueResult();

return Integer.parseInt(TransactionNo.toString());

}

@Override

public void deletePopularSearches() {

if(getTotalPopularSearches()>600) {

List<PopularSearch> ps = sessionFactory.getCurrentSession().createQuery("FROM PopularSearch").setMaxResults(10).list();

for (Iterator i = ps.iterator(); i.hasNext();) {

Object objPs = i.next();

sessionFactory.getCurrentSession().delete(objPs);

}

}

}

}

**StateDAO.txt**

package com.sharanamvaishali.dao;

import com.sharanamvaishali.model.State;

import java.util.List;

public interface StateDAO {

public void saveState(State state);

public List<State> listState();

public void removeState(Integer stateNo);

public State getStateById(Integer stateNo);

}

**StateDAOImpl.txt**

package com.sharanamvaishali.dao;

import com.sharanamvaishali.model.State;

import java.util.List;

import org.hibernate.SessionFactory;

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

import org.springframework.stereotype.Repository;

@Repository

public class StateDAOImpl implements StateDAO {

@Autowired

private SessionFactory sessionFactory;

@Override

public void saveState(State state) {

sessionFactory.getCurrentSession().merge(state);

}

@Override

public List<State> listState() {

return sessionFactory.getCurrentSession().createQuery("from State").list();

}

@Override

public void removeState(Integer stateNo) {

State state = (State) sessionFactory.getCurrentSession().load(State.class, stateNo);

if (null != state) {

sessionFactory.getCurrentSession().delete(state);

}

}

@Override

public State getStateById(Integer stateNo) {

return (State) sessionFactory.getCurrentSession().get(State.class, stateNo);

}

}

**TransactionDAO.txt**

package com.sharanamvaishali.dao;

import com.sharanamvaishali.model.Transaction;

import java.util.Date;

import java.util.List;

public interface TransactionDAO {

public void saveTransaction(Transaction transaction);

public Integer getNextTransactionNo();

public List<Transaction> listGroupedTransactions();

public List<Transaction> listTransactions();

public List<Transaction> listGroupedTransactionsByUsername(String userName);

public List<Transaction> listTransactionsByUsername(String userName);

public List<Transaction> listGroupedTransactionsByDate(Date date);

public List<Transaction> listTransactionsByDate(Date date);

public void updateGoogleOrderNo(Integer transactionNo, String orderNo);

}

**TransactionDAOImpl.txt**

package com.sharanamvaishali.dao;

import com.sharanamvaishali.model.Transaction;

import java.util.Date;

import java.util.List;

import org.hibernate.SessionFactory;

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

import org.springframework.stereotype.Repository;

@Repository

public class TransactionDAOImpl implements TransactionDAO {

@Autowired

private SessionFactory sessionFactory;

@Override

public void saveTransaction(Transaction transaction) {

sessionFactory.getCurrentSession().merge(transaction);

}

@Override

public Integer getNextTransactionNo() {

return Integer.parseInt(sessionFactory.getCurrentSession().createSQLQuery("SELECT MAX(TransactionNo)+1 AS NextTransactionNo FROM Transactions").list().get(0).toString());

}

@Override

public List<Transaction> listGroupedTransactions() {

return sessionFactory.getCurrentSession().createQuery("select transactionNo, transactionDate, userName, sum(qty\*cost), sum(qty), googleOrderNo from Transaction group by transactionNo").list();

}

@Override

public List<Transaction> listTransactions() {

return sessionFactory.getCurrentSession().createQuery("from Transaction").list();

}

@Override

public List<Transaction> listGroupedTransactionsByUsername(String userName) {

return sessionFactory.getCurrentSession().createQuery("select transactionNo, transactionDate, userName, sum(qty\*cost), sum(qty), googleOrderNo from Transaction where userName = '" + userName + "' group by transactionNo").list();

}

@Override

public List<Transaction> listTransactionsByUsername(String userName) {

return sessionFactory.getCurrentSession().createQuery("from Transaction where userName = '" + userName + "'").list();

}

@Override

public List<Transaction> listGroupedTransactionsByDate(Date date) {

return sessionFactory.getCurrentSession().createQuery("select transactionNo, transactionDate, userName, sum(qty\*cost), sum(qty), googleOrderNo from Transaction where date(transactionDate) = :date group by transactionNo")

.setDate("date", date)

.list();

}

@Override

public List<Transaction> listTransactionsByDate(Date date) {

return sessionFactory.getCurrentSession().createQuery("from Transaction where date(transactionDate) = :date")

.setDate("date", date)

.list();

}

@Override

public void updateGoogleOrderNo(Integer transactionNo, String orderNo) {

sessionFactory.getCurrentSession().createQuery("update Transaction set googleOrderNo = :orderNo where transactionNo = :transactionNo")

.setString("orderNo", orderNo)

.setInteger("transactionNo", transactionNo)

.executeUpdate();

}

}

**UserDAO.txt**

package com.sharanamvaishali.dao;

import com.sharanamvaishali.model.User;

import java.util.List;

public interface UserDAO {

public void saveUser(User User);

public List<User> listUser();

public void removeUser(Integer userNo);

public User getUserById(Integer userNo);

public User getUserByUserName(String userName);

}

**UserDAOImpl.txt**

package com.sharanamvaishali.dao;

import com.sharanamvaishali.model.User;

import java.util.List;

import org.hibernate.Query;

import org.hibernate.SessionFactory;

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

import org.springframework.stereotype.Repository;

@Repository

public class UserDAOImpl implements UserDAO {

@Autowired

private SessionFactory sessionFactory;

@Override

public void saveUser(User User) {

sessionFactory.getCurrentSession().merge(User);

}

@Override

public List<User> listUser() {

return sessionFactory.getCurrentSession().createQuery("from User").list();

}

@Override

public void removeUser(Integer userNo) {

User User = (User) sessionFactory.getCurrentSession().load(User.class, userNo);

if (null != User) {

sessionFactory.getCurrentSession().delete(User);

}

}

@Override

public User getUserById(Integer userNo) {

return (User) sessionFactory.getCurrentSession().get(User.class, userNo);

}

@Override

public User getUserByUserName(String userName) {

Query query = sessionFactory.getCurrentSession().createQuery("from User WHERE userName = :UserName");

query.setString("UserName", userName);

return (User) query.uniqueResult();

}

}