Library DAO Tutorial

Step 8: Hibernate Version – DAO Implementation

Step 8: Hibernate Version – DAO Implementation

Overview

In this document, we see how to implement the Library DAO using Hibernate and the Hibernate Session. The implementation of the DAO is fairly straightforward because we have already mapped all of the relationships in our data transfer objects.

Notes:

This class implements the HLibraryDao interface created in the previous step.

1. We use constructor injection to have Spring supply us with a Hibernate SessionFactory.
2. The private helper method **currentSession** is just a convenience wrapper method.
3. The getAll* methods require us to create a criteria based on the Class of the object we wish to retrieve. We then ask Hibernate to turn the result into a List.
4. We use the HBook class instead of the Book class.

```
package com.swcguild.library.dao;

import com.swcguild.library.model.Author;
import com.swcguild.library.model.HBook;
import com.swcguild.library.model.Publisher;
import java.util.List;
import javax.inject.Inject;
import org.hibernate.SessionFactory;
import org.hibernate.classic.Session;
import org.springframework.stereotype.Repository;
import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;

@Repository
@Transactional
public class LibraryDaoHibernateImpl implements HLibraryDao {

    private SessionFactory sessionFactory;

    @Inject
    public LibraryDaoHibernateImpl(SessionFactory sessionFactory) {
        this.sessionFactory = sessionFactory;
    }

    private Session currentSession() {
        return this.sessionFactory.getCurrentSession();
    }
```

```
@Override
@Transactional(propagation=Propagation.REQUIRED, readOnly=false)
public void addAuthor(Author author) {
    currentSession().save(author);
}

@Override
public void deleteAuthor(Author author) {
    currentSession().delete(author);
}

@Override
public void updateAuthor(Author author) {
    currentSession().update(author);
}

@Override
public Author getAuthorById(int id) {
    return (Author) currentSession().get(Author.class, id);
}

@Override
public List<Author> getAllAuthors() {
    return (List<Author>)currentSession().createCriteria(Author.class).list();
}

@Override
@Transactional(propagation=Propagation.REQUIRED, readOnly=false)
public void addBook(HBook book) {
    currentSession().save(book);
}

@Override
@Transactional(propagation=Propagation.REQUIRED, readOnly=false)
public void deleteBook(HBook book) {
    currentSession().delete(book);
}

@Override
@Transactional(propagation=Propagation.REQUIRED, readOnly=false)
public void updateBook(HBook book) {
    currentSession().update(book);
}
```

```
    @Override
    @Transactional(propagation=Propagation.REQUIRED, readOnly=true)
    public HBook getBookById(int id) {
        return (HBook) currentSession().get(HBook.class, id);
    }

    @Override
    @Transactional(propagation=Propagation.REQUIRED, readOnly=true)
    public List<HBook> getAllBooks() {
        return (List<HBook>) currentSession().createCriteria(HBook.class).list();
    }

    @Override
    @Transactional(propagation=Propagation.REQUIRED, readOnly=false)
    public void addPublisher(Publisher publisher) {
        currentSession().save(publisher);
    }

    @Override
    @Transactional(propagation=Propagation.REQUIRED, readOnly=false)
    public void deletePublisher(Publisher publisher) {
        currentSession().delete(publisher);
    }

    @Override
    @Transactional(propagation=Propagation.REQUIRED, readOnly=false)
    public void updatePublisher(Publisher publisher) {
        currentSession().update(publisher);
    }

    @Override
    @Transactional(propagation=Propagation.REQUIRED, readOnly=true)
    public Publisher getPublisherById(int id) {
        return (Publisher) currentSession().get(Publisher.class, id);
    }

    @Override
    @Transactional(propagation=Propagation.REQUIRED, readOnly=true)
    public List<Publisher> getAllPublishers() {
        return (List<Publisher>) currentSession().createCriteria(Publisher.class).list();
    }
}
```