

Lab 08

Session and Persistent Class

Mục tiêu

- Sử dụng SessionFactory của Hibernate để tương tác với database
- Tạo lớp persistent và thực hiện các chức năng của một ứng dụng web thông qua SessionFactory.

Phần I Bài tập step by step

Bài 1.1

Sử dụng Hibernate framework thực hiện các chức năng load và detail cho ứng dụng web spring mvc.

STEP 1: Tạo database cho bài tập này như sau:

```
create database SpringHibernate_Lab08
go
use SpringHibernate_Lab08
go
create table Categories(
    CategoryID varchar(10) not null primary key,
    CatetoryName varchar(30),
    Description varchar(100));

create table Products(
    ProductID varchar(10) not null primary key,
    CategoryID varchar(10),
    ProductName varchar(50),
    Price int,
    Description varchar(100)
    constraint fk_cat_pro foreign key (CategoryID) references Categories(CategoryID));

insert into Categories values('C0101', 'Beverages', 'Soft drinks, coffees, teas, beers, and ales');
insert into Categories values('C0102', 'Condiments', 'Sweet and savory sauces, relishes, spreads, and seasonings');
insert into Categories values('C0103', 'Confections', 'Desserts, candies, and sweet breads');
insert into Categories values('C0104', 'Dairy Products', 'Cheeses');
insert into Categories values('C0105', 'Grains_Cereals', 'Breads, crackers, pasta, and cereal');
insert into Categories values('C0106', 'Meat_Poultry', 'Prepared meats');
insert into Categories values('C0107', 'Produce', 'Dried fruit and bean curd');
insert into Categories values('C0108', 'Seafood', 'Seaweed and fish');
```

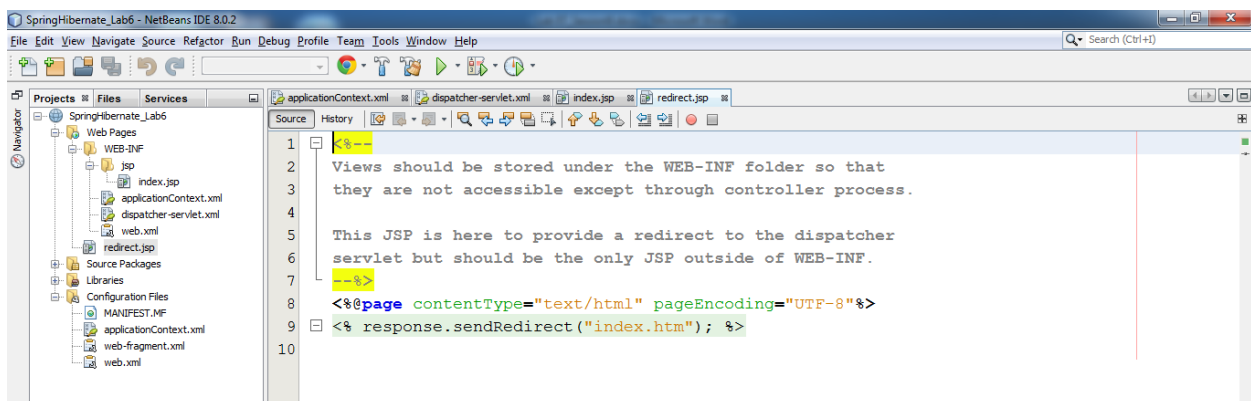
```
insert into Products values('P0101', 'C0101', 'Chai', 18, 'Good');
insert into Products values('P0102', 'C0101', 'Chang', 19, 'Good');
insert into Products values('P0103', 'C0101', 'Aniseed Syrup', 10, 'Good');
insert into Products values('P0104', 'C0102', 'Chef Anton's Cajun Seasoning', 22,
'Good');
insert into Products values('P0105', 'C0102', 'Chef Anton's Gumbo Mix', 21.35, 'Good');
insert into Products values('P0106', 'C0103', 'Grandma's Boysenberry Spread', 25,
'Good');
insert into Products values('P0107', 'C0103', 'Uncle Bob's Organic Dried Pears', 30,
'Good');
insert into Products values('P0108', 'C0103', 'Northwoods Cranberry Sauce', 40, 'Good');
insert into Products values('P0109', 'C0104', 'Mishi Kobe Niku', 97, 'Good');
insert into Products values('P01010', 'C0104', 'Ikura', 31, 'Good');
insert into Products values('P01011', 'C0105', 'Queso Cabrales', 21, 'Good');
insert into Products values('P01012', 'C0105', 'Queso Manchego La Pastora', 38, 'Good');

select * from Categories;
select * from Products;
```

STEP 2: Tạo một project web trong netbean đặt tên “SpringHibernate_Lab6”

New Project -> Java Web -> Web Application -> Đặt tên project “SpringHibernate_Lab6” ->

Next -> Next -> Tích chọn Spring Web MVC -> Finish



Add thư viện kết nối CSDL và thư viện của Hibernate vào ứng dụng

Right click Libraries -> Add Jar/Folder -> Tìm và add gói thư viện sqljdbc4.jar

Right click Libraries -> Add Library -> Add thư viện Hibernate (Hibernate 4.3.x(JPA2.1))

STEP 3: Tạo các class entity

Tạo class Category:

Source Package -> New -> Java Class.. -> Đặt tên class "Category", package "entities". Class này có các thuộc tính tương ứng với các cột của bảng Categories.

```
package entities;

import java.io.Serializable;
import java.util.Collection;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
/**
 *
 * @author HAITHANH
 */
@Entity
@Table(name = "Categories")
public class Category implements Serializable {

    @Id
    @Column(name = "CategoryID")
    private String categoryId;
    @Column(name = "CategoryName")
    private String categoryName;
    @Column(name = "Description")
    private String description;

    public Category() {
    }

    public Category(String categoryId, String categoryName, String description) {
        this.categoryId = categoryId;
        this.categoryName = categoryName;
        this.description = description;
    }

    public String getCategoryID() {
        return categoryId;
    }

    public void setCategoryID(String categoryId) {
        this.categoryId = categoryId;
    }
}
```

```
public String getCategoryName() {  
    return categoryName;  
}  
  
public void setCategoryName(String categoryName) {  
    this.categoryName = categoryName;  
}  
  
public String getDescription() {  
    return description;  
}  
  
public void setDescription(String description) {  
    this.description = description;  
}  
}
```

Tạo class Product:

entities -> New -> Java Class.. -> Đặt tên class "Product". Class này tương ứng với bảng Products.

```
package entities;  
  
import java.io.Serializable;  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.Id;  
import javax.persistence.Table;  
  
/**  
 *  
 * @author HAITHANH  
 */  
@Entity  
@Table(name = "Products")  
public class Product implements Serializable {  
    @Id  
    @Column(name = "ProductID")  
    private String productId;  
    @Column(name = "CategoryId")
```

```
private String categoryId;
@Column(name = "ProductName")
private String productName;
@Column(name = "Price")
private int price;
@Column(name = "Description")
private String description;

public Product() {
}

public Product(String productId, String categoryId, String productName, int price,
String description) {
    this.productId = productId;
    this.categoryId = categoryId;
    this.productName = productName;
    this.price = price;
    this.description = description;
}

public String getProductId() {
    return productId;
}

public void setProductId(String productId) {
    this.productId = productId;
}

public String getProductName() {
    return productName;
}

public void setProductName(String productName) {
    this.productName = productName;
}

public int getPrice() {
    return price;
}

public void setPrice(int price) {
    this.price = price;
}

public String getDescription() {
```

```
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    public String getCategoryId() {
        return categoryId;
    }

    public void setCategoryId(String categoryId) {
        this.categoryId = categoryId;
    }
}
```

STEP 4: Tạo giao diện và class DAO

Tạo interface CategoryDAO:

Source Packages -> New -> Other -> Java -> Java Interface -> Next -> Đặt tên interface “CategoryDAO”, package “dao”. Khai báo hàm để lấy về thông tin của bảng Categories

```
package dao;

import entities.Category;
import java.util.List;

/**
 *
 * @author HAITHANH
 */
public interface CategoryDAO {
    public List<Category> getAllCategories();
}
```

Tạo class CategoryDAOImpl thực thi giao diện CategoryDAO và cài đặt các hàm trong giao diện CategoryDAO:

dao -> New -> Java Class.. -> Đặt tên class "CategoryDAOImpl" -> Finish. Lớp này sử dụng đối tượng lớp SessionFatory của Hibernate để tương tác với database.

```
package dao;

import entities.Category;
import java.util.List;
import org.hibernate.Session;
import org.hibernate.SessionFactory;

/**
 *
 * @author HAITHANH
 */
public class CategoryDAOImpl implements CategoryDAO{

    private SessionFactory sessionFactory;

    public void setSessionFactory(SessionFactory sessionFactory) {
        this.sessionFactory = sessionFactory;
    }

    @Override
    public List<Category> getAllCategories() {
        Session session = sessionFactory.openSession();
        List<Category> list = session.createQuery("from Category").list();
        return list;
    }

}
```

Tạo giao diện ProductDAO và khai báo hàm để lấy về Product theo CategoryId:

dao -> New -> Other -> Java -> Java Interface -> Next -> Đặt tên interface "ProductDAO".

```
package dao;

import entities.Product;
import java.util.List;

/**
 *
```

```
* @author HAITHANH
*/
public interface ProductDAO {
    public List<Product> getProductByCateId(String cateId);
}
```

Tạo class ProductDAOImpl thực thi giao diện ProductDAO và cài đặt các hàm thực thi:

dao -> New -> Java Class.. -> Đặt tên class "ProductDAOImpl".

```
package dao;

import entities.Category;
import entities.Product;
import java.util.List;
import org.hibernate.Session;
import org.hibernate.SessionFactory;

/**
 *
 * @author HAITHANH
 */
public class ProductDAOImpl implements ProductDAO{
    private SessionFactory sessionFactory;

    public void setSessionFactory(SessionFactory sessionFactory) {
        this.sessionFactory = sessionFactory;
    }

    @Override
    public List<Product> getProductByCateId(String cateId) {
        Session session = sessionFactory.openSession();
        List<Product> list = session.createQuery("from Product where categoryId
= '"+cateId+"'").list();
        return list;
    }
}
```

STEP 5: Tạo class CategoryController

Spring/Hibernate Framework

Source Packages -> New -> Java Class.. -> Đặt tên class “CategoryController”, package “controller”. Thêm annotation và cài đặt hàm cho lớp này như sau:

```
package controller;

import dao.CategoryDAO;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

/**
 *
 * @author HAITHANH
 */
@Controller
public class CategoryController {
    private CategoryDAO categoryDAO;

    @Autowired
    @Qualifier(value = "categoryDAO")
    public void setCategoryDAO(CategoryDAO categoryDAO) {
        this.categoryDAO = categoryDAO;
    }

    @RequestMapping(value = "/loadCategories", method = RequestMethod.GET)
    public String loadCategories(Model model){
        model.addAttribute("listCate", categoryDAO.getAllCategories());
        return "index";
    }
}
```

STEP 6: Tạo class ProductController

controller -> New -> Java Class.. -> Đặt tên class “ProductController” Thêm annotation và cài đặt hàm cho lớp này như sau:

```
package controller;

import dao.ProductDAO;
```

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestParam;

/**
 *
 * @author HAITHANH
 */
@Controller
public class ProductController {
    private ProductDAO productDAO;

    @Autowired
    @Qualifier(value = "productDAO")
    public void setProductDAO(ProductDAO productDAO) {
        this.productDAO = productDAO;
    }

    @RequestMapping(value = "/loadProductByCategory")
    public String loadProductByCategory(@RequestParam("cateId")String cateId,
    Model model){
        model.addAttribute("listProduct", productDAO.getProductByCateId(cateId));
        return "listProduct";
    }
}
```

STEP 7: Cấu hình cho ứng dụng

Vào file “dispatcher-servlet.xml” để thêm cấu hình như sau:

```
<?xml version='1.0' encoding='UTF-8' ?>
<!-- was: <?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:p="http://www.springframework.org/schema/p"
    xmlns:aop="http://www.springframework.org/schema/aop"
    xmlns:tx="http://www.springframework.org/schema/tx"
    xmlns:mvc="http://www.springframework.org/schema/mvc"
    xmlns:context="http://www.springframework.org/schema/context"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-4.0.xsd
http://www.springframework.org/schema/aop
```

```
http://www.springframework.org/schema/aop/spring-aop-4.0.xsd
http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-4.0.xsd
http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd">

<mvc:annotation-driven></mvc:annotation-driven>
<context:component-scan base-package="controller"></context:component-scan>

<bean id="viewResolver"
    class="org.springframework.web.servlet.view.InternalResourceViewResolver"
    p:prefix="/WEB-INF/jsp/"
    p:suffix=".jsp" />

<bean id="dataSource"
class="org.springframework.jdbc.datasource.DriverManagerDataSource">
    <property name="driverClassName"
value="com.microsoft.sqlserver.jdbc.SQLServerDriver"/>
    <property name="url"
value="jdbc:sqlserver://localhost:1433;databaseName=SpringHibernate_Lab08"/>
    <property name="username" value="sa"/>
    <property name="password" value="1234$"/>
</bean>

<bean id="hibernate4AnnotatedSessionFactory"
class="org.springframework.orm.hibernate4.LocalSessionFactoryBean">
    <property name="dataSource" ref="dataSource"/>
    <property name="annotatedClasses">
        <list>
            <value>entities.Category</value>
            <value>entities.Product</value>
        </list>
    </property>
    <property name="hibernateProperties">
        <props>
            <prop
key="hibernate.dialect">org.hibernate.dialect.SQLServerDialect</prop>
            <prop key="hibernate.show_sql">true</prop>
        </props>
    </property>
</bean>
```

```
<bean id="categoryDAO" class="dao.CategoryDAOImpl">
  <property name="sessionFactory" ref="hibernate4AnnotatedSessionFactory"/>
</bean>

<bean id="productDAO" class="dao.ProductDAOImpl">
  <property name="sessionFactory" ref="hibernate4AnnotatedSessionFactory"/>
</bean>
</beans>
```

STEP 8: Thay đổi nội dung trang index.jsp để hiển thị thông tin các Categories

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>Welcome to Spring Web MVC project</title>
</head>

<body>
  <center>
    <h1>LIST CATEGORIES</h1>
    <table border="1">
      <tr>
        <th>Category Id</th>
        <th>Category Name</th>
        <th>Description</th>
        <th>Product Details</th>
      </tr>
      <c:forEach items="${listCate}" var="c">
        <tr>
          <td>${c.categoryId}</td>
          <td>${c.categoryName}</td>
          <td>${c.description}</td>
          <td>
            <a
href="loadProductByCategory.htm?cateId=${c.categoryId}&cateName=${c.categoryName}">list products</a>
          </td>
        </tr>
      </c:forEach>
```

```
</table>
</center>
</body>
</html>
```

Tạo trang listProduct.jsp để hiển thị thông tin các sản phẩm theo Category:

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>JSP Page</title>
</head>
<body>
<center>
<h1>LIST PRODUCT OF CATEGORY: <b>${cateName}</b></h1>
<table border="1">
<tr>
<th>Product Id</th>
<th>Product Name</th>
<th>Category Name</th>
<th>Price</th>
<th>Description</th>
</tr>
<c:forEach items="${listProduct}" var="p">
<tr>
<td>${p.productId}</td>
<td>${p.productName}</td>
<td>${cateName}</td>
<td>${p.price}</td>
<td>${p.description}</td>
</tr>
</c:forEach>
</table>
<a href="redirect.jsp">Back</a>
</center>
</body>
</html>
```

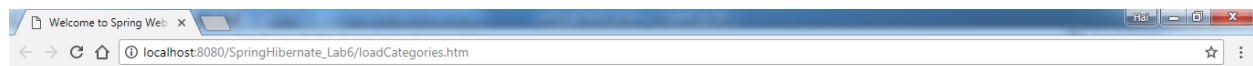
STEP 9: Đổi điều hướng trang redirect.jsp và chạy thử ứng dụng.

Vào trang “redirect.jsp” đổi điều hướng thành:

```
<% response.sendRedirect("loadCategories.htm"); %>
```

Chạy thử ứng dụng:

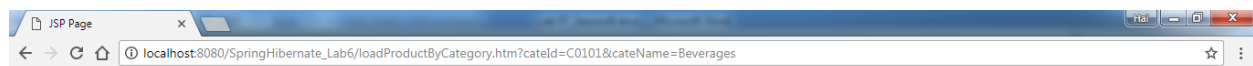
Trang index.jsp khi ứng dụng được chạy lần đầu tiên:



LIST CATEGORIES

Category Id	Category Name	Description	Product Details
C0101	Beverages	Soft drinks, coffees, teas, beers, and ales	list products
C0102	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings	list products
C0103	Confections	Desserts, candies, and sweet breads	list products
C0104	Dairy Products	Cheeses	list products
C0105	Grains_Cereals	Breads, crackers, pasta, and cereal	list products
C0106	Meat_Poultry	Prepared meats	list products
C0107	Produce	Dried fruit and bean curd	list products
C0108	Seafood	Seaweed and fish	list products

Trang listProduct.jsp khi click vào link Product Details:



LIST PRODUCT OF CATEGORY: Beverages

Product Id	Product Name	Category Name	Price	Description
P0101	Chai	Beverages	18	Good
P0102	Chang	Beverages	19	Good
P0103	Aniseed Syrup	Beverages	10	Good

[Back](#)

Phần bài tập làm thêm:

Bài 1

Tạo database như sau:

Tên database: DB_SpringHibernate_HomeWork_Lab7

Tên bảng: TblEmployees

Tên cột	Kiểu dữ liệu	Mô tả
EmpId	int – Tự tăng	Mã nhân viên - Khóa chính
EmpName	nvarchar(50)	Tên nhân viên
Department	nvarchar(50)	Tên phòng ban
BasicSalary	float	Lương cơ bản
Bonus	float	Thưởng

Tạo ứng dụng SpringMVC kết hợp Hibernate để tạo các trang web có các chức năng sau:

- Load all: Hiển thị thông tin tất cả nhân viên khi ứng dụng được chạy lần đầu tiên
- Detail: Hiển thị chi tiết thông tin của một nhân viên
- Insert: Thêm mới 1 nhân viên
- Update: Cập nhật thông tin của 1 nhân viên đã có
- Delete: Xóa thông tin của 1 nhân viên.