Факультет компьютерных технологий

Практическое задание №2

«Современные платформы программирования»

Вариант №6

Выполнил: студент гр. 981064 Ефименко Павел Викторович

Проверил: Бакунов А. М.

Минск 2021

Условие:

Реализовать WEB приложение и использование фреймворка Struts2, задействовать action. Обработать ошибки. Связать компоненты с помощью фреймворка Spring.

Код:

Db

DROP TABLE IF EXISTS `mkyong`.`customer`;

CREATE TABLE `mkyong`.`customer` (

`CUSTOMER\_ID` bigint(20) unsigned NOT NULL AUTO\_INCREMENT,

`NAME` varchar(45) NOT NULL,

`ADDRESS` varchar(255) NOT NULL,

`CREATED\_DATE` datetime NOT NULL,

PRIMARY KEY (`CUSTOMER\_ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=17 DEFAULT CHARSET=utf8;

Pom.xml

*<?*xml version="1.0" encoding="UTF-8"*?>*<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <groupId>org.ardlema</groupId>  
 <artifactId>cookbook-struts2</artifactId>  
 <version>1.0-SNAPSHOT</version>  
 <packaging>war</packaging>  
 <name>cookbook</name>  
  
 <properties>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
 <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>  
 <servlet.port>8080</servlet.port>  
  
 <struts2.version>2.5.26</struts2.version>  
 <spring.version>5.3.12</spring.version>  
 <hibernate.version>5.6.1.Final</hibernate.version>  
 <sl4j.version>1.7.2</sl4j.version>  
 <hsqldb.version>2.2.9</hsqldb.version>  
 <mysql.connector.java.version>8.0.25</mysql.connector.java.version>  
 </properties>  
  
 <repositories>  
 <repository>  
 <id>java.net</id>  
 <url>http://download.java.net/maven/2/</url>  
 </repository>  
 </repositories>  
  
  
 <dependencies>  
  
 <dependency>  
 <groupId>org.apache.struts</groupId>  
 <artifactId>struts2-core</artifactId>  
 <version>${struts2.version}</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.apache.struts</groupId>  
 <artifactId>struts2-config-browser-plugin</artifactId>  
 <version>${struts2.version}</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.apache.struts</groupId>  
 <artifactId>struts2-junit-plugin</artifactId>  
 <version>${struts2.version}</version>  
 <scope>test</scope>  
 </dependency>  
  
 <dependency>  
 <groupId>commons-logging</groupId>  
 <artifactId>commons-logging</artifactId>  
 <version>1.2</version>  
 </dependency>  
  
 <dependency>  
 <groupId>junit</groupId>  
 <artifactId>junit</artifactId>  
 <version>4.13.2</version>  
 <scope>test</scope>  
 </dependency>  
  
 <dependency>  
 <groupId>org.hamcrest</groupId>  
 <artifactId>hamcrest-all</artifactId>  
 <version>1.3</version>  
 </dependency>  
  
  
 <dependency>  
 <groupId>javax.servlet</groupId>  
 <artifactId>servlet-api</artifactId>  
 <version>2.5</version>  
 <scope>provided</scope>  
 </dependency>  
  
 <dependency>  
 <groupId>javax.servlet</groupId>  
 <artifactId>jsp-api</artifactId>  
 <version>2.0</version>  
 <scope>provided</scope>  
 </dependency>  
  
 <dependency>  
 <groupId>org.seleniumhq.selenium</groupId>  
 <artifactId>selenium-java</artifactId>  
 <version>4.0.0</version>  
 </dependency>  
  
 *<!-- Spring 3 dependencies -->* <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-core</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-tx</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-jdbc</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-orm</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
  
  
 *<!-- Struts 2 + Spring 3 need this jar, ContextLoaderListener -->* <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-web</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
  
  
  
 *<!-- Struts 2 + Spring integration plugins -->* <dependency>  
 <groupId>org.apache.struts</groupId>  
 <artifactId>struts2-spring-plugin</artifactId>  
 <version>2.5.26</version>  
 </dependency>  
  
 *<!-- Mockito -->* <dependency>  
 <groupId>org.mockito</groupId>  
 <artifactId>mockito-all</artifactId>  
 <version>1.10.19</version>  
 </dependency>  
  
 *<!-- Hibernate -->* <dependency>  
 <groupId>org.hibernate</groupId>  
 <artifactId>hibernate-core</artifactId>  
 <version>${hibernate.version}</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.javassist</groupId>  
 <artifactId>javassist</artifactId>  
 <version>3.28.0-GA</version>  
 </dependency>  
  
 <dependency>  
 <groupId>mysql</groupId>  
 <artifactId>mysql-connector-java</artifactId>  
 <version>${mysql.connector.java.version}</version>  
 <scope>runtime</scope>  
 </dependency>  
  
 <dependency>  
 <groupId>dom4j</groupId>  
 <artifactId>dom4j</artifactId>  
 <version>20040902.021138</version>  
 </dependency>  
  
 *<!-- h2 database -->* <dependency>  
 <groupId>com.h2database</groupId>  
 <artifactId>h2</artifactId>  
 <version>1.4.200</version>  
 </dependency>  
  
 </dependencies>  
  
 <build>  
 <pluginManagement>  
 <plugins>  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-surefire-plugin</artifactId>  
 <version>2.8</version>  
 <configuration>  
 <additionalClasspathElements>  
 <additionalClasspathElement>test/resources</additionalClasspathElement>  
 </additionalClasspathElements>  
 </configuration>  
 </plugin>  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-failsafe-plugin</artifactId>  
 <version>2.8</version>  
 </plugin>  
 <plugin>  
 <groupId>org.codehaus.cargo</groupId>  
 <artifactId>cargo-maven2-plugin</artifactId>  
 <version>1.4.2</version>  
 <configuration>  
 <configuration>  
 <properties>  
 <cargo.servlet.port>${servlet.port}</cargo.servlet.port>  
 </properties>  
 </configuration>  
 <deployables>  
 *<!--  
 The project's artifact is automatically deployed if no deployable  
 is defined. However, we define it here so that we can specify the  
 context (we don't want the version to be included in the context).  
 -->* <deployable>  
 <groupId>${project.groupId}</groupId>  
 <artifactId>${project.artifactId}</artifactId>  
 <type>war</type>  
 <properties>  
 <context>${project.artifactId}</context>  
 </properties>  
 </deployable>  
 </deployables>  
 </configuration>  
 </plugin>  
  
  
 <plugin>  
 <groupId>org.codehaus.mojo</groupId>  
 <artifactId>sql-maven-plugin</artifactId>  
  
 <dependencies>  
 *<!-- specify the dependent jdbc driver here -->* <dependency>  
 <groupId>mysql</groupId>  
 <artifactId>mysql-connector-java</artifactId>  
 <version>${mysql.connector.java.version}</version>  
 </dependency>  
 </dependencies>  
  
 <configuration>  
 <driver>org.gjt.mm.mysql.Driver</driver>  
 <url>jdbc:mysql://localhost/cookbook</url>  
 <username>root</username>  
 <password>root</password>  
 <srcFiles>  
 <srcFile>src/test/resources/schema.sql</srcFile>  
 <srcFile>src/test/resources/test-data.sql</srcFile>  
 </srcFiles>  
 *<!--all executions are ignored if -Dmaven.test.skip=true-->* <skip>${maven.test.skip}</skip>  
  
 </configuration>  
  
 <executions>  
  
 *<!-- It creates integration test data before running the tests -->* <execution>  
 <id>create-integration-test-data</id>  
 <phase>process-test-resources</phase>  
 <goals>  
 <goal>execute</goal>  
 </goals>  
 <configuration>  
 <orderFile>ascending</orderFile>  
 <fileset>  
 <basedir>/</basedir>  
 <includes>  
 <include>schema.sql</include>  
 <include>test-data.sql</include>  
 </includes>  
 </fileset>  
 </configuration>  
 </execution>  
 </executions>  
  
 </plugin>  
  
  
  
 </plugins>  
 </pluginManagement>  
 <plugins>  
 <plugin>  
 <groupId>org.mortbay.jetty</groupId>  
 <artifactId>jetty-maven-plugin</artifactId>  
 <version>8.1.2.v20120308</version>  
 <configuration>  
 <scanintervalseconds>2</scanintervalseconds>  
 </configuration>  
 </plugin>  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-failsafe-plugin</artifactId>  
 <executions>  
 <execution>  
 <id>perform-integration</id>  
 <goals>  
 <goal>integration-test</goal>  
 </goals>  
 <configuration>  
 <systemProperties>  
 <property>  
 <name>servlet.port</name>  
 <value>${servlet.port}</value>  
 </property>  
 </systemProperties>  
 </configuration>  
 </execution>  
 <execution>  
 <id>verify-integration</id>  
 <goals>  
 <goal>verify</goal>  
 </goals>  
 </execution>  
 </executions>  
 </plugin>  
 <plugin>  
 <groupId>org.codehaus.cargo</groupId>  
 <artifactId>cargo-maven2-plugin</artifactId>  
 <executions>  
 <execution>  
 <id>start-cargo</id>  
 <phase>pre-integration-test</phase>  
 <goals>  
 <goal>start</goal>  
 </goals>  
 </execution>  
 <execution>  
 <id>stop-cargo</id>  
 <phase>post-integration-test</phase>  
 <goals>  
 <goal>stop</goal>  
 </goals>  
 </execution>  
 </executions>  
 </plugin>  
 </plugins>  
 </build>  
</project>

Customer.java

import java.util.Date;

public class Customer implements java.io.Serializable {

private Long customerId;

private String name;

private String address;

private Date createdDate;

//getter and setter methods

}

Customer.hbm.xml

<?xml version="1.0"?>

<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">

<!-- Generated 20 Julai 2010 11:40:18 AM by Hibernate Tools 3.2.5.Beta -->

<hibernate-mapping>

<class name="com.mkyong.customer.model.Customer"

table="customer" catalog="mkyong">

<id name="customerId" type="java.lang.Long">

<column name="CUSTOMER\_ID" />

<generator class="identity" />

</id>

<property name="name" type="string">

<column name="NAME" length="45" not-null="true" />

</property>

<property name="address" type="string">

<column name="ADDRESS" not-null="true" />

</property>

<property name="createdDate" type="timestamp">

<column name="CREATED\_DATE" length="19" not-null="true" />

</property>

</class>

</hibernate-mapping>

CustomerBo.java

import java.util.List;

import com.mkyong.customer.model.Customer;

public interface CustomerBo{

void addCustomer(Customer customer);

List<Customer> listCustomer();

}

CustomerBoImpl.java

import java.util.List;

import com.mkyong.customer.bo.CustomerBo;

import com.mkyong.customer.dao.CustomerDAO;

import com.mkyong.customer.model.Customer;

public class CustomerBoImpl implements CustomerBo{

CustomerDAO customerDAO;

//DI via Spring

public void setCustomerDAO(CustomerDAO customerDAO) {

this.customerDAO = customerDAO;

}

//call DAO to save customer

public void addCustomer(Customer customer){

customerDAO.addCustomer(customer);

}

//call DAO to return customers

public List<Customer> listCustomer(){

return customerDAO.listCustomer();

}

}

CustomerDAO.java

import java.util.List;

import com.mkyong.customer.model.Customer;

public interface CustomerDAO{

void addCustomer(Customer customer);

List<Customer> listCustomer();

}

**CustomerDAOImpl.java**

import java.util.List;

import org.springframework.orm.hibernate3.support.HibernateDaoSupport;

import com.mkyong.customer.dao.CustomerDAO;

import com.mkyong.customer.model.Customer;

public class CustomerDAOImpl extends HibernateDaoSupport

implements CustomerDAO{

//add the customer

public void addCustomer(Customer customer){

getHibernateTemplate().save(customer);

}

//return all the customers in list

public List<Customer> listCustomer(){

return getHibernateTemplate().find("from Customer");

}

}

**CustomerAction.java**

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

import com.mkyong.customer.bo.CustomerBo;

import com.mkyong.customer.model.Customer;

import com.opensymphony.xwork2.ModelDriven;

public class CustomerAction implements ModelDriven{

Customer customer = new Customer();

List<Customer> customerList = new ArrayList<Customer>();

CustomerBo customerBo;

//DI via Spring

public void setCustomerBo(CustomerBo customerBo) {

this.customerBo = customerBo;

}

public Object getModel() {

return customer;

}

public List<Customer> getCustomerList() {

return customerList;

}

public void setCustomerList(List<Customer> customerList) {

this.customerList = customerList;

}

//save customer

public String addCustomer() throws Exception{

//save it

customer.setCreatedDate(new Date());

customerBo.addCustomer(customer);

//reload the customer list

customerList = null;

customerList = customerBo.listCustomer();

return "success";

}

//list all customers

public String listCustomer() throws Exception{

customerList = customerBo.listCustomer();

return "success";

}

}

CustomerBean.xml

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

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-2.5.xsd">

<bean id="customerAction" class="com.mkyong.customer.action.CustomerAction">

<property name="customerBo" ref="customerBo" />

</bean>

<bean id="customerBo" class="com.mkyong.customer.bo.impl.CustomerBoImpl" >

<property name="customerDAO" ref="customerDAO" />

</bean>

<bean id="customerDAO" class="com.mkyong.customer.dao.impl.CustomerDAOImpl" >

<property name="sessionFactory" ref="sessionFactory" />

</bean>

</beans>

database.properties

jdbc.driverClassName=com.mysql.jdbc.Driver

jdbc.url=jdbc:mysql://localhost:3306/mpp\_labs

jdbc.username=mysql

jdbc.password= mysql

DataSource.xml

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-2.5.xsd">

<bean

class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">

<property name="location">

<value>WEB-INF/classes/config/database/properties/database.properties</value>

</property>

</bean>

<bean id="dataSource"

class="org.springframework.jdbc.datasource.DriverManagerDataSource">

<property name="driverClassName" value="${jdbc.driverClassName}" />

<property name="url" value="${jdbc.url}" />

<property name="username" value="${jdbc.username}" />

<property name="password" value="${jdbc.password}" />

</bean>

</beans>

HibernateSessionFactory.xml

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

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-2.5.xsd">

<!-- Hibernate session factory -->

<bean id="sessionFactory"

class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">

<property name="dataSource">

<ref bean="dataSource"/>

</property>

<property name="hibernateProperties">

<props>

<prop key="hibernate.dialect">org.hibernate.dialect.MySQLDialect</prop>

<prop key="hibernate.show\_sql">true</prop>

</props>

</property>

<property name="mappingResources">

<list>

<value>com/mkyong/customer/hibernate/Customer.hbm.xml</value>

</list>

</property>

</bean>

</beans>

SpringBeans.xml

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-2.5.xsd">

<!-- Database Configuration -->

<import resource="config/spring/DataSource.xml"/>

<import resource="config/spring/HibernateSessionFactory.xml"/>

<!-- Beans Declaration -->

<import resource="com/mkyong/customer/spring/CustomerBean.xml"/>

</beans>

**customer.jsp**

<%@ taglib prefix="s" uri="/struts-tags" %>

<html>

<head>

</head>

<body>

<h2>Add Customer</h2>

<s:form action="addCustomerAction" >

<s:textfield name="name" label="Name" value="" />

<s:textarea name="address" label="Address" value="" cols="50" rows="5" />

<s:submit />

</s:form>

<h2>All Customers</h2>

<s:if test="customerList.size() > 0">

<table border="1px" cellpadding="8px">

<tr>

<th>Customer Id</th>

<th>Name</th>

<th>Address</th>

<th>Created Date</th>

</tr>

<s:iterator value="customerList" status="userStatus">

<tr>

<td><s:property value="customerId" /></td>

<td><s:property value="name" /></td>

<td><s:property value="address" /></td>

<td><s:date name="createdDate" format="dd/MM/yyyy" /></td>

</tr>

</s:iterator>

</table>

</s:if>

<br/>

<br/>

</body>

</html>

struts.xml

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

<!DOCTYPE struts PUBLIC

"-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"

"http://struts.apache.org/dtds/struts-2.0.dtd">

<struts>

<constant name="struts.devMode" value="true" />

<package name="default" namespace="/" extends="struts-default">

<action name="addCustomerAction"

class="customerAction" method="addCustomer" >

<result name="success">pages/customer.jsp</result>

</action>

<action name="listCustomerAction"

class="customerAction" method="listCustomer" >

<result name="success">pages/customer.jsp</result>

</action>

</package>

</struts>

**web.xml**

<!DOCTYPE web-app PUBLIC

"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"

"http://java.sun.com/dtd/web-app\_2\_3.dtd" >

<web-app>

<display-name>Struts 2 Web Application</display-name>

<filter>

<filter-name>struts2</filter-name>

<filter-class>

org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter

</filter-class>

</filter>

<filter-mapping>

<filter-name>struts2</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>/WEB-INF/classes/SpringBeans.xml</param-value>

</context-param>

<listener>

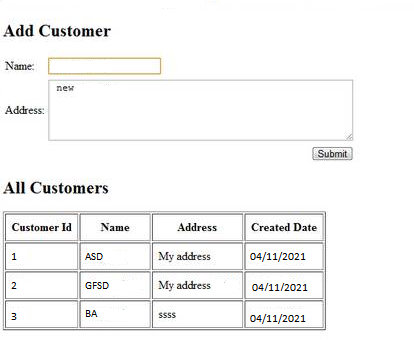
<listener-class>

org.springframework.web.context.ContextLoaderListener

</listener-class>

</listener>

</web-app>

Результат кода: ****

Вывод:

Были изучены возможности создания приложения с использованием Struts2 и Spring фреймворков на языке программирования Java.