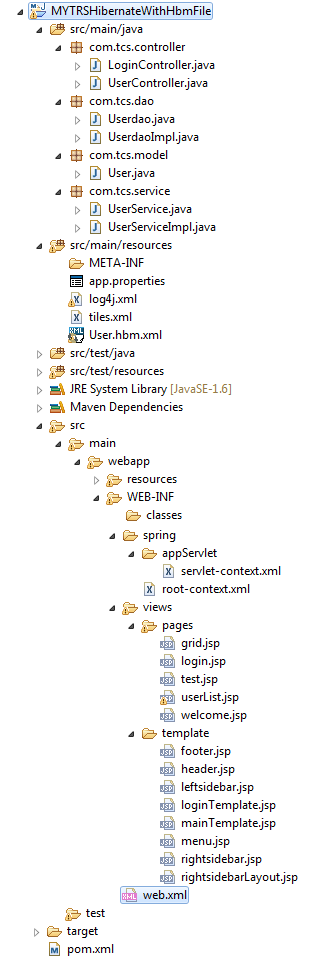
* Spring Hibernate Integration

1. **Using XML Mapping Configuration**

**Project Structure:**

****

**Pom.xml:**

<dependencies>

<!-- Spring -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${org.springframework-version}</version>

<exclusions>

<!-- Exclude Commons Logging in favor of SLF4j -->

<exclusion>

<groupId>commons-logging</groupId>

<artifactId>commons-logging</artifactId>

</exclusion>

</exclusions>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjrt</artifactId>

<version>${org.aspectj-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<!-- AspectJ -->

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjrt</artifactId>

<version>${org.aspectj-version}</version>

</dependency>

<!-- Logging -->

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>${org.slf4j-version}</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>jcl-over-slf4j</artifactId>

<version>${org.slf4j-version}</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-log4j12</artifactId>

<version>${org.slf4j-version}</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>log4j</groupId>

<artifactId>log4j</artifactId>

<version>1.2.15</version>

<exclusions>

<exclusion>

<groupId>javax.mail</groupId>

<artifactId>mail</artifactId>

</exclusion>

<exclusion>

<groupId>javax.jms</groupId>

<artifactId>jms</artifactId>

</exclusion>

<exclusion>

<groupId>com.sun.jdmk</groupId>

<artifactId>jmxtools</artifactId>

</exclusion>

<exclusion>

<groupId>com.sun.jmx</groupId>

<artifactId>jmxri</artifactId>

</exclusion>

</exclusions>

<scope>runtime</scope>

</dependency>

<!-- @Inject -->

<dependency>

<groupId>javax.inject</groupId>

<artifactId>javax.inject</artifactId>

<version>1</version>

</dependency>

<!-- Servlet -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>servlet-api</artifactId>

<version>2.5</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>javax.servlet.jsp</groupId>

<artifactId>jsp-api</artifactId>

<version>2.1</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

<!-- Test -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.7</version>

<scope>test</scope>

</dependency>

<!-- MySQL database driver -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.1.9</version>

</dependency>

<!-- Hibernate framework -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>4.2.4.Final</version>

</dependency>

<!-- Tiles dependencies -->

<dependency>

<groupId>org.apache.tiles</groupId>

<artifactId>tiles-api</artifactId>

<version>2.2.2</version>

</dependency>

<dependency>

<groupId>org.apache.tiles</groupId>

<artifactId>tiles-autotag-core-runtime</artifactId>

<version>1.1.0</version>

</dependency>

<dependency>

<groupId>org.apache.tiles</groupId>

<artifactId>tiles-core</artifactId>

<version>2.2.2</version>

</dependency>

<dependency>

<groupId>org.apache.tiles</groupId>

<artifactId>tiles-jsp</artifactId>

<version>2.2.2</version>

</dependency>

<dependency>

<groupId>org.apache.tiles</groupId>

<artifactId>tiles-servlet</artifactId>

<version>2.2.2</version>

</dependency>

<dependency>

<groupId>org.apache.tiles</groupId>

<artifactId>tiles-template</artifactId>

<version>2.2.2</version>

</dependency>

</dependencies>

**Web.xml:**

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<web-appxmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*xmlns=*"http://java.sun.com/xml/ns/javaee"*xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"*version=*"2.5"*>

<context-param>

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

<param-value>/WEB-INF/spring/root-context.xml</param-value>

</context-param>

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

</listener>

<servlet>

<servlet-name>appServlet</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<init-param>

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

<param-value>/WEB-INF/spring/appServlet/servlet-context.xml</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>appServlet</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

</web-app>

**Servlet-context.xml:**

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<beans:beansxmlns=*"http://www.springframework.org/schema/mvc"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*xmlns:beans=*"http://www.springframework.org/schema/beans"*

xmlns:p=*"http://www.springframework.org/schema/p"*

xmlns:context=*"http://www.springframework.org/schema/context"*xmlns:tx=*"http://www.springframework.org/schema/tx"*

xsi:schemaLocation=*"http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc.xsd*

*http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/tx*

*http://www.springframework.org/schema/tx/spring-tx.xsd*

*http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd"*>

<annotation-driven/>

<resourcesmapping=*"/resources/\*\*"*location=*"/resources/mytheme/"*/>

<context:property-placeholderlocation=*"classpath:app.properties"*/>

<context:component-scanbase-package=*"com.tcs"*/>

<beans:beanid=*"dataSource"*

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

<beans:propertyname=*"driverClassName"*value=*"${app.jdbc.driverClassName}"*/>

<beans:propertyname=*"url"*value=*"${app.jdbc.url}"*/>

<beans:propertyname=*"username"*value=*"${app.jdbc.username}"*/>

<beans:propertyname=*"password"*value=*"${app.jdbc.password}"*/>

</beans:bean>

<beans:beanid=*"sessionFactory"*

class=*"org.springframework.orm.hibernate4.LocalSessionFactoryBean"*>

<beans:propertyname=*"dataSource"*>

<beans:refbean=*"dataSource"*/>

</beans:property>

<beans:propertyname=*"packagesToScan"*value=*"com.tcs.model"*>

</beans:property>

<beans:propertyname=*"hibernateProperties"*>

<beans:props>

<beans:propkey=*"hibernate.hbm2ddl.auto"*>update

</beans:prop>

<beans:propkey=*"hibernate.show\_sql"*>true

</beans:prop>

<beans:propkey=*"hibernate.dialect"*>

org.hibernate.dialect.MySQLDialect

</beans:prop>

</beans:props>

</beans:property>

<beans:propertyname=*"mappingResources"*>

<beans:list>

<beans:value>User.hbm.xml</beans:value>

</beans:list>

</beans:property>

</beans:bean>

<beans:beanid=*"transactionManager"*

class=*"org.springframework.orm.hibernate4.HibernateTransactionManager"*>

<beans:propertyname=*"sessionFactory"*>

<beans:refbean=*"sessionFactory"*/>

</beans:property>

</beans:bean>

<tx:annotation-driventransaction-manager=*"transactionManager"*/>

<!-- Tiles Configuration -->

<beans:beanid=*"tilesConfigurer"*

class=*"org.springframework.web.servlet.view.tiles2.TilesConfigurer"*

p:definitions=*"classpath:tiles.xml"*/>

<beans:beanid=*"tilesViewResolver"*

class=*"org.springframework.web.servlet.view.UrlBasedViewResolver"*

p:viewClass=*"org.springframework.web.servlet.view.tiles2.TilesView"*/>

</beans:beans>

**App.properties:**

app.jdbc.driverClassName=com.mysql.jdbc.Driver

app.jdbc.url=jdbc:mysql://localhost:3306/mybatis

app.jdbc.username=root

app.jdbc.password=

**tiles.xml:**

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<!DOCTYPEtiles-definitionsPUBLIC

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

"http://tiles.apache.org/dtds/tiles-config\_2\_0.dtd">

<tiles-definitions>

<definitionname=*"loginTemplate"*template=*"/WEB-INF/views/template/loginTemplate.jsp"*>

<put-attributename=*"body"*value=*""*></put-attribute>

</definition>

<definitionextends=*"loginTemplate"*name=*"login"*>

<put-attributename=*"title"*value=*"Login :: MyTRS"*></put-attribute>

<put-attributename=*"body"*value=*"/WEB-INF/views/pages/login.jsp"*></put-attribute>

</definition>

<definitionname=*"base.definition"*template=*"/WEB-INF/views/template/mainTemplate.jsp"*>

<put-attributename=*"title"*value=*""*></put-attribute>

<put-attributename=*"header"*value=*"/WEB-INF/views/template/header.jsp"*></put-attribute>

<put-attributename=*"menu"*value=*"/WEB-INF/views/template/menu.jsp"*></put-attribute>

<put-attributename=*"body"*value=*""*></put-attribute>

<put-attributename=*"sidebar"*value=*"/WEB-INF/views/template/leftsidebar.jsp"*></put-attribute>

<put-attributename=*"footer"*value=*"/WEB-INF/views/template/footer.jsp"*></put-attribute>

</definition>

<definitionextends=*"base.definition"*name=*"welcome"*>

<put-attributename=*"title"*value=*"Dashboard"*></put-attribute>

<put-attributename=*"body"*value=*"/WEB-INF/views/pages/welcome.jsp"*></put-attribute>

</definition>

<definitionextends=*"base.definition"*name=*"userList"*>

<put-attributename=*"title"*value=*"List Of Users"*></put-attribute>

<put-attributename=*"body"*value=*"/WEB-INF/views/pages/userList.jsp"*></put-attribute>

</definition>

<definitionextends=*"base.definition"*name=*"edit"*>

<put-attributename=*"title"*value=*"Edit User Details"*></put-attribute>

<put-attributename=*"body"*value=*"/WEB-INF/views/pages/edit.jsp"*></put-attribute>

</definition>

<definitionextends=*"base.definition"*name=*"grid"*>

<put-attributename=*"title"*value=*"List Of Users"*></put-attribute>

<put-attributename=*"body"*value=*"/WEB-INF/views/pages/grid.jsp"*></put-attribute>

</definition>

<definitionname=*"rightsidebarLayout"*template=*"/WEB-INF/views/template/rightsidebarLayout.jsp"*>

<put-attributename=*"title"*value=*""*></put-attribute>

<put-attributename=*"header"*value=*"/WEB-INF/views/template/header.jsp"*></put-attribute>

<put-attributename=*"menu"*value=*"/WEB-INF/views/template/menu.jsp"*></put-attribute>

<put-attributename=*"body"*value=*""*></put-attribute>

<put-attributename=*"rightsidebar"*value=*"/WEB-INF/views/template/rightsidebar.jsp"*></put-attribute>

<put-attributename=*"footer"*value=*"/WEB-INF/views/template/footer.jsp"*></put-attribute>

</definition>

<definitionextends=*"rightsidebarLayout"*name=*"test"*>

<put-attributename=*"title"*value=*"List Of Users"*></put-attribute>

<put-attributename=*"body"*value=*"/WEB-INF/views/pages/test.jsp"*></put-attribute>

</definition>

</tiles-definitions>

**User.hbm.xml:**

<?xmlversion=*"1.0"*encoding=*"utf-8"*?>

<!DOCTYPEhibernate-mappingPUBLIC

"-//Hibernate/Hibernate Mapping DTD//EN"

"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping>

<classname=*"com.tcs.model.User"*table=*"MYUSER"*>

<metaattribute=*"class-description"*>

This class contains the User detail.

</meta>

<idname=*"user\_id"*type=*"int"*column=*"USER\_ID"*>

<generatorclass=*"native"*/>

</id>

<propertyname=*"username"*column=*"USERNAME"*type=*"string"*/>

<propertyname=*"password"*column=*"PASSWORD"*type=*"string"*/>

<propertyname=*"firstName"*column=*"FIRSTNAME"*type=*"string"*/>

<propertyname=*"lastName"*column=*"LASTNAME"*type=*"string"*/>

<propertyname=*"phone"*column=*"PHONE"*type=*"string"*/>

<propertyname=*"emailId"*column=*"EMAILID"*type=*"string"*/>

</class>

</hibernate-mapping>

**Log4j.xml:**

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<!DOCTYPElog4j:configurationSYSTEM"log4j.dtd">

<log4j:configurationdebug=*"true"*

xmlns:log4j=*'http://jakarta.apache.org/log4j/'*>

<!-- Console -->

<appendername=*"console"*class=*"org.apache.log4j.ConsoleAppender"*>

<layoutclass=*"org.apache.log4j.PatternLayout"*>

<paramname=*"ConversionPattern"*

value=*"%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L - %m%n"*/>

</layout>

</appender>

<!-- file -->

<appendername=*"file"*class=*"org.apache.log4j.RollingFileAppender"*>

<paramname=*"append"*value=*"false"*/>

<paramname=*"maxFileSize"*value=*"10KB"*/>

<paramname=*"maxBackupIndex"*value=*"5"*/>

<paramname=*"file"*value=*"F:/Coding/SpringCoding/WAE/logs/logs.log"*/>

<layoutclass=*"org.apache.log4j.PatternLayout"*>

<paramname=*"ConversionPattern"*

value=*"%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L - %m%n"*/>

</layout>

</appender>

<root>

<levelvalue=*"DEBUG"*/>

<appender-refref=*"console"*/>

<appender-refref=*"file"*/>

</root>

</log4j:configuration>

**User.java:**

**package** com.tcs.model;

**import** javax.persistence.Transient;

**publicclass** User {

**privateint**user\_id;

**private** String username;

**private** String password;

**private** String firstName;

**private** String lastName;

**private** String phone;

**private** String emailId;

**privateboolean**valid;

@Transient

**publicboolean** isValid() {

**return**valid;

}

**publicvoid** setValid(**boolean** valid) {

**this**.valid = valid;

}

**publicint** getUser\_id() {

**return**user\_id;

}

**publicvoid** setUser\_id(**int** user\_id) {

**this**.user\_id = user\_id;

}

**public** String getUsername() {

**return**username;

}

**publicvoid** setUsername(String username) {

**this**.username = username;

}

**public** String getPassword() {

**return**password;

}

**publicvoid** setPassword(String password) {

**this**.password = password;

}

**public** String getFirstName() {

**return**firstName;

}

**publicvoid** setFirstName(String firstName) {

**this**.firstName = firstName;

}

**public** String getLastName() {

**return**lastName;

}

**publicvoid** setLastName(String lastName) {

**this**.lastName = lastName;

}

**public** String getPhone() {

**return**phone;

}

**publicvoid** setPhone(String phone) {

**this**.phone = phone;

}

**public** String getEmailId() {

**return**emailId;

}

**publicvoid** setEmailId(String emailId) {

**this**.emailId = emailId;

}

@Override

**public** String toString() {

**return**"Id: " + user\_id + "; fname : " + firstName + "' phone : " + phone + "; emailId: " + emailId;

}

}

**UserController.java:**

**package** com.tcs.controller;

**import** java.util.List;

**import** java.util.Locale;

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

**import** org.springframework.stereotype.Controller;

**import** org.springframework.ui.Model;

**import** org.springframework.validation.BindingResult;

**import** org.springframework.web.bind.annotation.ModelAttribute;

**import** org.springframework.web.bind.annotation.PathVariable;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestMethod;

**import** com.tcs.model.User;

**import** com.tcs.service.UserService;

@Controller

**publicclass** UserController {

@Autowired

UserService service;

@RequestMapping(value = "/save", method = RequestMethod.*POST*)

**public** String save(@ModelAttribute("user") User user,

BindingResult br, Model model) {

**if** (br.hasErrors()) {

**return**"welcome";

}

**if** (user.getUser\_id() > 0) {

service.updateUser(user);

model.addAttribute("showUpdateMessage", "true");

} **else** {

service.addUser(user);

model.addAttribute("showSaveMessage", "true");

}

**return**"redirect:userList";

}

@RequestMapping(value = { "userList"}, method = RequestMethod.*GET*)

**public** String list(Model model) {

List<User> userlist = service.listUser();

model.addAttribute("userlist", userlist);

**return**"userList";

}

@RequestMapping(value = "/edit/{id}", method = RequestMethod.*GET*)

**public** String edit(@PathVariable Integer id, Model model) {

model.addAttribute("showUpdateMessage", "true");

model.addAttribute("user", service.getUser(id));

**return**"welcome";

}

@RequestMapping(value = "/delete/{id}", method = RequestMethod.*GET*)

**public** String delete(@PathVariable Integer id,Model model) {

service.deleteUser(id);

model.addAttribute("showDeleteMessage", "true");

**return**"redirect:/userList";

}

}

**UserService.java:**

**package** com.tcs.service;

**import** java.util.List;

**import** com.tcs.model.User;

**publicinterface** UserService {

**public** User getUser(**int** id);

**publicvoid** updateUser(User person);

**public** List<User> listUser();

**publicvoid** deleteUser(**int** id);

**publicvoid** addUser(User person);

**public** User validateUser(User usr);

}

**UserServiceImpl.java:**

**package** com.tcs.service;

**import** java.util.List;

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

**import** org.springframework.stereotype.Service;

**import** org.springframework.transaction.annotation.Transactional;

**import** com.tcs.dao.Userdao;

**import** com.tcs.model.User;

@Service

@Transactional

**publicclass** UserServiceImpl **implements** UserService {

@Autowired

Userdao userDao;

@Override

**public** User getUser(**int** id) {

**return**userDao.getUser(id);

}

@Override

**publicvoid** updateUser(User user) {

userDao.updateUser(user);

}

@Override

**public** List<User> listUser() {

**return**userDao.listUsers();

}

@Override

**publicvoid** deleteUser(**int** id) {

userDao.deleteUser(id);

}

@Override

**publicvoid** addUser(User user) {

userDao.addUser(user);

}

@Override

**public** User validateUser(User usr) {

**return**userDao.validateUser(usr);

}

}

**UserDao.java:**

**package** com.tcs.dao;

**import** java.util.List;

**import** com.tcs.model.User;

**publicinterface** Userdao {

**public** User getUser(**int** id);

**public** List<User> listUsers();

**publicvoid** deleteUser(**int** id);

**publicvoid** addUser(User user);

**publicvoid** updateUser(User user);

**public** User validateUser(User user);

}

**UserDaoImpl.java:**

**package** com.tcs.dao;

**import** java.util.List;

**import** org.hibernate.Query;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

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

**import** org.springframework.stereotype.Repository;

**import** com.tcs.model.User;

@Repository

**publicclass** UserdaoImpl **implements** Userdao {

@Autowired

**private** SessionFactory sessionFactory;

**public** Session getCurrentSession() {

**return**sessionFactory.getCurrentSession();

}

@Override

**public** User getUser(**int** id) {

User user = (User) getCurrentSession().get(User.**class**, id);

**return** user;

}

@SuppressWarnings("unchecked")

@Override

**public** List<User> listUsers() {

Query query = getCurrentSession().createQuery("from User");

List<User> list = query.list();

**return** list;

}

@Override

**publicvoid** deleteUser(**int** id) {

User user = getUser(id);

getCurrentSession().delete(user);

}

@Override

**publicvoid** addUser(User user) {

getCurrentSession().save(user);

}

@Override

**publicvoid** updateUser(User user) {

getCurrentSession().update(user);

}

@SuppressWarnings("rawtypes")

@Override

**public** User validateUser(User usr) {

Query query = getCurrentSession().createQuery("from User where username= :username and password= :password");

query.setParameter("username",usr.getUsername());

query.setParameter("password",usr.getPassword());

List list = query.list();

**if** (list.size() > 0) {

usr = (User) list.get(0);

usr.setValid(**true**);

} **else** {

usr.setValid(**false**);

}

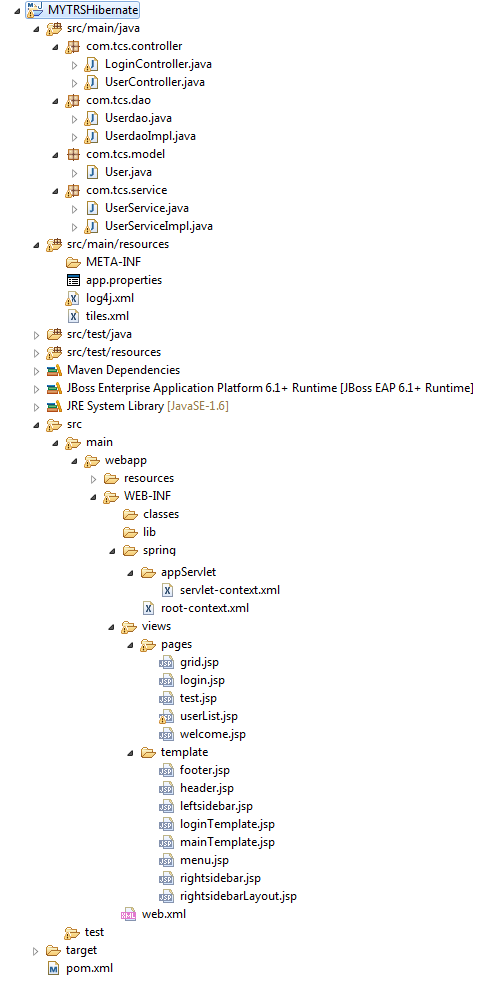
**return** usr;

}

}

1. **Using Annotations**

**Project Structure:**

****

**Pom.xml:**

<dependencies>

<!-- Spring -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${org.springframework-version}</version>

<exclusions>

<!-- Exclude Commons Logging in favor of SLF4j -->

<exclusion>

<groupId>commons-logging</groupId>

<artifactId>commons-logging</artifactId>

</exclusion>

</exclusions>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjrt</artifactId>

<version>${org.aspectj-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<!-- AspectJ -->

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjrt</artifactId>

<version>${org.aspectj-version}</version>

</dependency>

<!-- Logging -->

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>${org.slf4j-version}</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>jcl-over-slf4j</artifactId>

<version>${org.slf4j-version}</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-log4j12</artifactId>

<version>${org.slf4j-version}</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>log4j</groupId>

<artifactId>log4j</artifactId>

<version>1.2.15</version>

<exclusions>

<exclusion>

<groupId>javax.mail</groupId>

<artifactId>mail</artifactId>

</exclusion>

<exclusion>

<groupId>javax.jms</groupId>

<artifactId>jms</artifactId>

</exclusion>

<exclusion>

<groupId>com.sun.jdmk</groupId>

<artifactId>jmxtools</artifactId>

</exclusion>

<exclusion>

<groupId>com.sun.jmx</groupId>

<artifactId>jmxri</artifactId>

</exclusion>

</exclusions>

<scope>runtime</scope>

</dependency>

<!-- @Inject -->

<dependency>

<groupId>javax.inject</groupId>

<artifactId>javax.inject</artifactId>

<version>1</version>

</dependency>

<!-- Servlet -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>servlet-api</artifactId>

<version>2.5</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>javax.servlet.jsp</groupId>

<artifactId>jsp-api</artifactId>

<version>2.1</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

<!-- Test -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.7</version>

<scope>test</scope>

</dependency>

<!-- MySQL database driver -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.1.9</version>

</dependency>

<!-- Hibernate framework -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>4.2.4.Final</version>

</dependency>

<!-- Tiles dependencies -->

<dependency>

<groupId>org.apache.tiles</groupId>

<artifactId>tiles-api</artifactId>

<version>2.2.2</version>

</dependency>

<dependency>

<groupId>org.apache.tiles</groupId>

<artifactId>tiles-autotag-core-runtime</artifactId>

<version>1.1.0</version>

</dependency>

<dependency>

<groupId>org.apache.tiles</groupId>

<artifactId>tiles-core</artifactId>

<version>2.2.2</version>

</dependency>

<dependency>

<groupId>org.apache.tiles</groupId>

<artifactId>tiles-jsp</artifactId>

<version>2.2.2</version>

</dependency>

<dependency>

<groupId>org.apache.tiles</groupId>

<artifactId>tiles-servlet</artifactId>

<version>2.2.2</version>

</dependency>

<dependency>

<groupId>org.apache.tiles</groupId>

<artifactId>tiles-template</artifactId>

<version>2.2.2</version>

</dependency>

</dependencies>

**Web.xml:**

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<web-appxmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*xmlns=*"http://java.sun.com/xml/ns/javaee"*xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"*version=*"2.5"*>

<context-param>

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

<param-value>/WEB-INF/spring/root-context.xml</param-value>

</context-param>

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

</listener>

<servlet>

<servlet-name>appServlet</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<init-param>

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

<param-value>/WEB-INF/spring/appServlet/servlet-context.xml</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>appServlet</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

</web-app>

**Servlet-context.xml:**

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<beans:beansxmlns=*"http://www.springframework.org/schema/mvc"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*xmlns:beans=*"http://www.springframework.org/schema/beans"*

xmlns:p=*"http://www.springframework.org/schema/p"*

xmlns:context=*"http://www.springframework.org/schema/context"*xmlns:tx=*"http://www.springframework.org/schema/tx"*

xsi:schemaLocation=*"http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc.xsd*

*http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/tx*

*http://www.springframework.org/schema/tx/spring-tx.xsd*

*http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd"*>

<annotation-driven/>

<resourcesmapping=*"/resources/\*\*"*location=*"/resources/mytheme/"*/>

<context:property-placeholderlocation=*"classpath:app.properties"*/>

<context:component-scanbase-package=*"com.tcs"*/>

<beans:beanid=*"dataSource"*

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

<beans:propertyname=*"driverClassName"*value=*"${app.jdbc.driverClassName}"*/>

<beans:propertyname=*"url"*value=*"${app.jdbc.url}"*/>

<beans:propertyname=*"username"*value=*"${app.jdbc.username}"*/>

<beans:propertyname=*"password"*value=*"${app.jdbc.password}"*/>

</beans:bean>

<beans:beanid=*"sessionFactory"*

class=*"org.springframework.orm.hibernate4.LocalSessionFactoryBean"*>

<beans:propertyname=*"dataSource"*>

<beans:refbean=*"dataSource"*/>

</beans:property>

<beans:propertyname=*"packagesToScan"*value=*"com.tcs.model"*>

</beans:property>

<beans:propertyname=*"hibernateProperties"*>

<beans:props>

<beans:propkey=*"hibernate.hbm2ddl.auto"*>

update

</beans:prop>

<beans:propkey=*"hibernate.show\_sql"*>

true

</beans:prop>

<beans:propkey=*"hibernate.dialect"*>

org.hibernate.dialect.MySQLDialect

</beans:prop>

</beans:props>

</beans:property>

</beans:bean>

<beans:beanid=*"transactionManager"*

class=*"org.springframework.orm.hibernate4.HibernateTransactionManager"*>

<beans:propertyname=*"sessionFactory"*>

<beans:refbean=*"sessionFactory"*/>

</beans:property>

</beans:bean>

<tx:annotation-driventransaction-manager=*"transactionManager"*/>

<!-- Tiles Configuration -->

<beans:beanid=*"tilesConfigurer"*

class=*"org.springframework.web.servlet.view.tiles2.TilesConfigurer"*

p:definitions=*"classpath:tiles.xml"*/>

<beans:beanid=*"tilesViewResolver"*

class=*"org.springframework.web.servlet.view.UrlBasedViewResolver"*

p:viewClass=*"org.springframework.web.servlet.view.tiles2.TilesView"*/>

</beans:beans>

**App.properties:**

app.jdbc.driverClassName=com.mysql.jdbc.Driver

app.jdbc.url=jdbc:mysql://localhost:3306/mybatis

app.jdbc.username=root

app.jdbc.password=

**tiles.xml:**

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<!DOCTYPEtiles-definitionsPUBLIC

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

"http://tiles.apache.org/dtds/tiles-config\_2\_0.dtd">

<tiles-definitions>

<definitionname=*"loginTemplate"*template=*"/WEB-INF/views/template/loginTemplate.jsp"*>

<put-attributename=*"body"*value=*""*></put-attribute>

</definition>

<definitionextends=*"loginTemplate"*name=*"login"*>

<put-attributename=*"title"*value=*"Login :: MyTRS"*></put-attribute>

<put-attributename=*"body"*value=*"/WEB-INF/views/pages/login.jsp"*>

</put-attribute>

</definition>

<definitionname=*"base.definition"*template=*"/WEB-INF/views/template/mainTemplate.jsp"*>

<put-attributename=*"title"*value=*""*></put-attribute>

<put-attributename=*"header"*value=*"/WEB-INF/views/template/header.jsp"*></put-attribute>

<put-attributename=*"menu"*value=*"/WEB-INF/views/template/menu.jsp"*>

</put-attribute>

<put-attributename=*"body"*value=*""*></put-attribute>

<put-attributename=*"sidebar"*value=*"/WEB-INF/views/template/leftsidebar.jsp"*></put-attribute>

<put-attributename=*"footer"*value=*"/WEB-INF/views/template/footer.jsp"*></put-attribute>

</definition>

<definitionextends=*"base.definition"*name=*"welcome"*>

<put-attributename=*"title"*value=*"Dashboard"*></put-attribute>

<put-attributename=*"body"*value=*"/WEB-INF/views/pages/welcome.jsp"*></put-attribute>

</definition>

<definitionextends=*"base.definition"*name=*"userList"*>

<put-attributename=*"title"*value=*"List Of Users"*></put-attribute>

<put-attributename=*"body"*value=*"/WEB-INF/views/pages/userList.jsp"*></put-attribute>

</definition>

<definitionextends=*"base.definition"*name=*"edit"*>

<put-attributename=*"title"*value=*"Edit User Details"*></put-attribute>

<put-attributename=*"body"*value=*"/WEB-INF/views/pages/edit.jsp"*></put-attribute>

</definition>

<definitionextends=*"base.definition"*name=*"grid"*>

<put-attributename=*"title"*value=*"List Of Users"*></put-attribute>

<put-attributename=*"body"*value=*"/WEB-INF/views/pages/grid.jsp"*>

</put-attribute>

</definition>

<definitionname=*"rightsidebarLayout"*template=*"/WEB-INF/views/template/rightsidebarLayout.jsp"*>

<put-attributename=*"title"*value=*""*></put-attribute>

<put-attributename=*"header"*value=*"/WEB-INF/views/template/header.jsp"*></put-attribute>

<put-attributename=*"menu"*value=*"/WEB-INF/views/template/menu.jsp"*></put-attribute>

<put-attributename=*"body"*value=*""*></put-attribute>

<put-attributename=*"rightsidebar"*value=*"/WEB-INF/views/template/rightsidebar.jsp"*></put-attribute>

<put-attributename=*"footer"*value=*"/WEB-INF/views/template/footer.jsp"*></put-attribute>

</definition>

<definitionextends=*"rightsidebarLayout"*name=*"test"*>

<put-attributename=*"title"*value=*"List Of Users"*></put-attribute>

<put-attributename=*"body"*value=*"/WEB-INF/views/pages/test.jsp"*></put-attribute>

</definition>

</tiles-definitions>

**Log4j.xml:**

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<!DOCTYPElog4j:configurationSYSTEM"log4j.dtd">

<log4j:configurationdebug=*"true"*

xmlns:log4j=*'http://jakarta.apache.org/log4j/'*>

<!-- Console -->

<appendername=*"console"*class=*"org.apache.log4j.ConsoleAppender"*>

<layoutclass=*"org.apache.log4j.PatternLayout"*>

<paramname=*"ConversionPattern"*

value=*"%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L - %m%n"*/>

</layout>

</appender>

<!-- file -->

<appendername=*"file"*class=*"org.apache.log4j.RollingFileAppender"*>

<paramname=*"append"*value=*"false"*/>

<paramname=*"maxFileSize"*value=*"10KB"*/>

<paramname=*"maxBackupIndex"*value=*"5"*/>

<paramname=*"file"*value=*"F:/Coding/SpringCoding/WAE/logs/logs.log"*/>

<layoutclass=*"org.apache.log4j.PatternLayout"*>

<paramname=*"ConversionPattern"*

value=*"%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L - %m%n"*/>

</layout>

</appender>

<root>

<levelvalue=*"DEBUG"*/>

<appender-refref=*"console"*/>

<appender-refref=*"file"*/>

</root>

</log4j:configuration>

**User.java:**

**package** com.tcs.model;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.Table;

**import** javax.persistence.Transient;

@Entity

@Table(name = "myuser")

**publicclass** User {

@Id

@GeneratedValue(strategy = GenerationType.*IDENTITY*)

@Column(unique = **true**, nullable = **false**)

**privateint**user\_id;

@Column

String username;

@Column

String password;

@Column

String firstName;

@Column

String lastName;

@Column

String phone;

@Column

String emailId;

**privateboolean**valid;

@Transient

**publicboolean** isValid() {

**return**valid;

}

**publicvoid** setValid(**boolean** valid) {

**this**.valid = valid;

}

**publicint** getUser\_id() {

**return**user\_id;

}

**publicvoid** setUser\_id(**int** user\_id) {

**this**.user\_id = user\_id;

}

**public** String getUsername() {

**return**username;

}

**publicvoid** setUsername(String username) {

**this**.username = username;

}

**public** String getPassword() {

**return**password;

}

**publicvoid** setPassword(String password) {

**this**.password = password;

}

**public** String getFirstName() {

**return**firstName;

}

**publicvoid** setFirstName(String firstName) {

**this**.firstName = firstName;

}

**public** String getLastName() {

**return**lastName;

}

**publicvoid** setLastName(String lastName) {

**this**.lastName = lastName;

}

**public** String getPhone() {

**return**phone;

}

**publicvoid** setPhone(String phone) {

**this**.phone = phone;

}

**public** String getEmailId() {

**return**emailId;

}

**publicvoid** setEmailId(String emailId) {

**this**.emailId = emailId;

}

@Override

**public** String toString() {

**return**"Id: " + user\_id + "; fname : " + firstName + "' phone : "

+ phone + "; emailId: " + emailId;

}

}

**UserController.java:**

**package** com.tcs.controller;

**import** java.util.List;

**import** java.util.Locale;

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

**import** org.springframework.stereotype.Controller;

**import** org.springframework.ui.Model;

**import** org.springframework.validation.BindingResult;

**import** org.springframework.web.bind.annotation.ModelAttribute;

**import** org.springframework.web.bind.annotation.PathVariable;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestMethod;

**import** com.tcs.model.User;

**import** com.tcs.service.UserService;

@Controller

**publicclass** UserController {

@Autowired

UserService service;

@RequestMapping(value = "/save", method = RequestMethod.*POST*)

**public** String save(@ModelAttribute("user") User user,

BindingResult br, Model model) {

**if** (br.hasErrors()) {

**return**"welcome";

}

**if** (user.getUser\_id() > 0) {

service.updateUser(user);

model.addAttribute("showUpdateMessage", "true");

} **else** {

service.addUser(user);

model.addAttribute("showSaveMessage", "true");

}

**return**"redirect:userList";

}

@RequestMapping(value = { "userList"}, method = RequestMethod.*GET*)

**public** String list(Model model) {

List<User> userlist = service.listUser();

model.addAttribute("userlist", userlist);

**return**"userList";

}

@RequestMapping(value = "/edit/{id}", method = RequestMethod.*GET*)

**public** String edit(@PathVariable Integer id, Model model) {

model.addAttribute("showUpdateMessage", "true");

model.addAttribute("user", service.getUser(id));

**return**"welcome";

}

@RequestMapping(value = "/delete/{id}", method = RequestMethod.*GET*)

**public** String delete(@PathVariable Integer id,Model model) {

service.deleteUser(id);

model.addAttribute("showDeleteMessage", "true");

**return**"redirect:/userList";

}

@RequestMapping(value = "grid", method = RequestMethod.*GET*)

**public** String showGrid(Locale locale, Model model) {

List<User> users = service.listUser();

model.addAttribute("users", users);

**return**"grid";

}

}

**UserService.java:**

**package** com.tcs.service;

**import** java.util.List;

**import** com.tcs.model.User;

**publicinterface** UserService {

**public** User getUser(**int** id);

**publicvoid** updateUser(User user);

**public** List<User> listUser();

**publicvoid** deleteUser(**int** id);

**publicvoid** addUser(User user);

**public** User validateUser(User usr);

}

**UserServiceImpl.java:**

**package** com.tcs.service;

**import** java.util.List;

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

**import** org.springframework.stereotype.Service;

**import** org.springframework.transaction.annotation.Transactional;

**import** com.tcs.dao.Userdao;

**import** com.tcs.model.User;

@Service

@Transactional

**publicclass** UserServiceImpl **implements** UserService {

@Autowired

Userdao userDao;

@Override

**public** User getUser(**int** id) {

**return**userDao.getUser(id);

}

@Override

**publicvoid** updateUser(User user) {

userDao.updateUser(user);

}

@Override

**public** List<User> listUser() {

**return**userDao.listUsers();

}

@Override

**publicvoid** deleteUser(**int** id) {

userDao.deleteUser(id);

}

@Override

**publicvoid** addUser(User user) {

userDao.addUser(user);

}

@Override

**public** User validateUser(User usr) {

**return**userDao.validateUser(usr);

}

}

**UserDao.java:**

**package** com.tcs.dao;

**import** java.util.List;

**import** com.tcs.model.User;

**publicinterface** Userdao {

**public** User getUser(**int** id);

**public** List<User> listUsers();

**publicvoid** deleteUser(**int** id);

**publicvoid** addUser(User user);

**publicvoid** updateUser(User user);

**public** User validateUser(User user);

}

**UserDaoImpl.java:**

**package** com.tcs.dao;

**import** java.util.List;

**import** org.hibernate.Query;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

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

**import** org.springframework.stereotype.Repository;

**import** com.tcs.model.User;

@Repository

**publicclass** UserdaoImpl **implements** Userdao {

@Autowired

**private** SessionFactory sessionFactory;

**public** Session getCurrentSession() {

**return**sessionFactory.getCurrentSession();

}

@Override

**public** User getUser(**int** id) {

User user = (User) getCurrentSession().get(User.**class**, id);

**return** user;

}

@Override

**public** List<User> listUsers() {

**return**getCurrentSession().createQuery("From User").list();

}

@Override

**publicvoid** deleteUser(**int** id) {

User user = getUser(id);

getCurrentSession().delete(user);

}

@Override

**publicvoid** addUser(User user) {

getCurrentSession().save(user);

}

@Override

**publicvoid** updateUser(User user) {

getCurrentSession().update(user);

}

@Override

**public** User validateUser(User usr) {

Query query = getCurrentSession().createQuery("from User where username= :username and password= :password");

query.setParameter("username",usr.getUsername());

query.setParameter("password",usr.getPassword());

List list = query.list();

**if** (list.size() > 0) {

usr = (User) list.get(0);

usr.setValid(**true**);

} **else** {

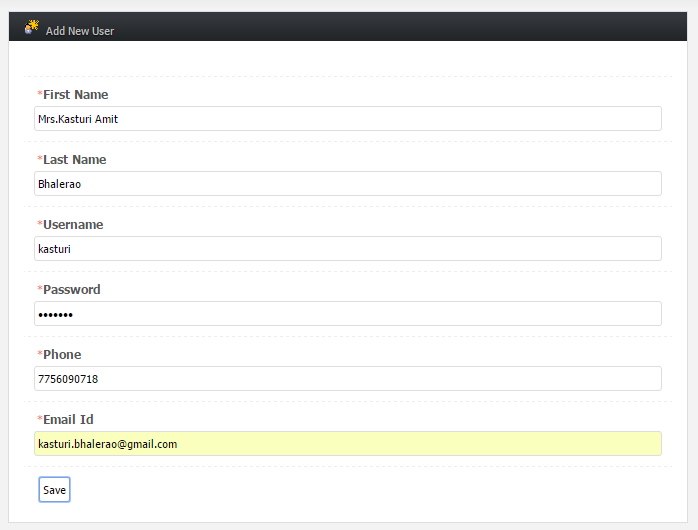
usr.setValid(**false**);

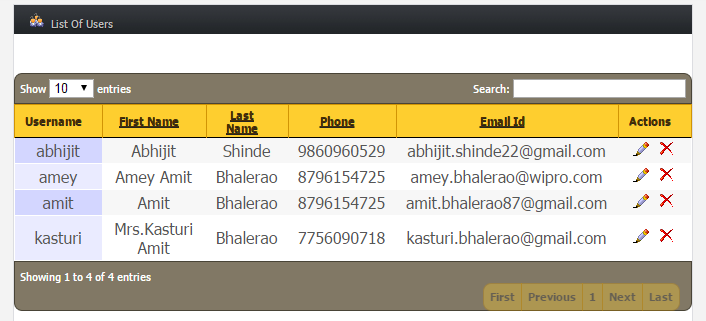
}

**return** usr;

}

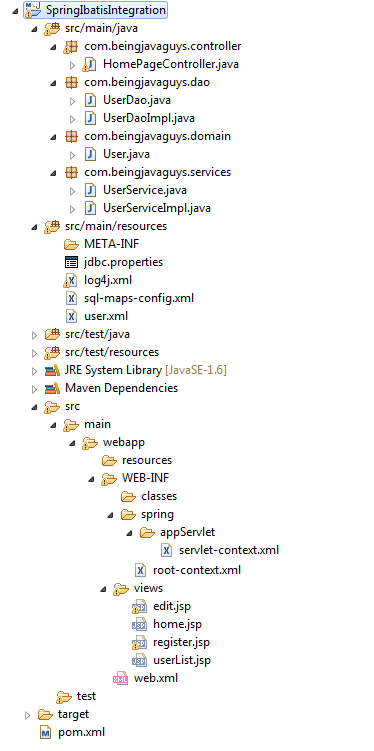
}





* Spring iBatis Integration

**Project Structure:**

****

**Pom.xml:**

<dependencies>

<!-- Spring -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${org.springframework-version}</version>

<exclusions>

<!-- Exclude Commons Logging in favor of SLF4j -->

<exclusion>

<groupId>commons-logging</groupId>

<artifactId>commons-logging</artifactId>

</exclusion>

</exclusions>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjrt</artifactId>

<version>${org.aspectj-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<!-- AspectJ -->

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjrt</artifactId>

<version>${org.aspectj-version}</version>

</dependency>

<!-- Logging -->

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>${org.slf4j-version}</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>jcl-over-slf4j</artifactId>

<version>${org.slf4j-version}</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-log4j12</artifactId>

<version>${org.slf4j-version}</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>log4j</groupId>

<artifactId>log4j</artifactId>

<version>1.2.15</version>

<exclusions>

<exclusion>

<groupId>javax.mail</groupId>

<artifactId>mail</artifactId>

</exclusion>

<exclusion>

<groupId>javax.jms</groupId>

<artifactId>jms</artifactId>

</exclusion>

<exclusion>

<groupId>com.sun.jdmk</groupId>

<artifactId>jmxtools</artifactId>

</exclusion>

<exclusion>

<groupId>com.sun.jmx</groupId>

<artifactId>jmxri</artifactId>

</exclusion>

</exclusions>

<scope>runtime</scope>

</dependency>

<!-- @Inject -->

<dependency>

<groupId>javax.inject</groupId>

<artifactId>javax.inject</artifactId>

<version>1</version>

</dependency>

<!-- Servlet -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>servlet-api</artifactId>

<version>2.5</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>javax.servlet.jsp</groupId>

<artifactId>jsp-api</artifactId>

<version>2.1</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

<!-- Test -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.7</version>

<scope>test</scope>

</dependency>

<!-- MySQL database driver -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.1.9</version>

</dependency>

<dependency>

<groupId>commons-dbcp</groupId>

<artifactId>commons-dbcp</artifactId>

<version>1.4</version>

</dependency>

<!—iBatis Framework -->

<dependency>

<groupId>org.apache.ibatis</groupId>

<artifactId>ibatis-sqlmap</artifactId>

<version>2.3.4.726</version>

</dependency>

<!-- Tiles dependencies -->

<dependency>

<groupId>org.apache.tiles</groupId>

<artifactId>tiles-api</artifactId>

<version>2.2.2</version>

</dependency>

<dependency>

<groupId>org.apache.tiles</groupId>

<artifactId>tiles-autotag-core-runtime</artifactId>

<version>1.1.0</version>

</dependency>

<dependency>

<groupId>org.apache.tiles</groupId>

<artifactId>tiles-core</artifactId>

<version>2.2.2</version>

</dependency>

<dependency>

<groupId>org.apache.tiles</groupId>

<artifactId>tiles-jsp</artifactId>

<version>2.2.2</version>

</dependency>

<dependency>

<groupId>org.apache.tiles</groupId>

<artifactId>tiles-servlet</artifactId>

<version>2.2.2</version>

</dependency>

<dependency>

<groupId>org.apache.tiles</groupId>

<artifactId>tiles-template</artifactId>

<version>2.2.2</version>

</dependency>

</dependencies>

**Web.xml:**

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<web-appversion=*"2.5"*xmlns=*"http://java.sun.com/xml/ns/javaee"*

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

xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"*>

<context-param>

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

<param-value>/WEB-INF/spring/root-context.xml</param-value>

</context-param>

<listener>

<listener-class>

org.springframework.web.context.ContextLoaderListener

</listener-class>

</listener>

<servlet>

<servlet-name>appServlet</servlet-name>

<servlet-class>

org.springframework.web.servlet.DispatcherServlet

</servlet-class>

<init-param>

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

<param-value>

/WEB-INF/spring/appServlet/servlet-context.xml

</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>appServlet</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

</web-app>

**Servlet-context.xml:**

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<beans:beansxmlns=*"http://www.springframework.org/schema/mvc"*

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

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

xmlns:context=*"http://www.springframework.org/schema/context"*

xsi:schemaLocation=*"http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc.xsd*

*http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd"*>

<annotation-driven/>

<context:property-placeholderlocation=*"classpath:jdbc.properties"*/>

<beans:beanclass=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

<beans:propertyname=*"prefix"*value=*"/WEB-INF/views/"*/>

<beans:propertyname=*"suffix"*value=*".jsp"*/>

</beans:bean>

<context:component-scanbase-package=*"com.beingjavaguys.controller"*/>

<beans:beanid=*"userDao"*class=*"com.beingjavaguys.dao.UserDaoImpl"*/>

<beans:beanid=*"userService"*class=*"com.beingjavaguys.services.UserServiceImpl"*/>

<beans:beanid=*"propertyConfigurer"*class=*"org.springframework.beans.factory.config.PropertyPlaceholderConfigurer"*>

<beans:propertyname=*"locations"*>

<beans:list>

<beans:value>classpath:jdbc.properties</beans:value>

</beans:list>

</beans:property>

</beans:bean>

<beans:beanid=*"dataSource1"*class=*"org.apache.commons.dbcp.BasicDataSource"*

destroy-method=*"close"*>

<beans:propertyname=*"driverClassName"*value=*"${app.jdbc.driverClassName}"*/>

<beans:propertyname=*"url"*value=*"${app.jdbc.url}"*/>

<beans:propertyname=*"username"*value=*"${app.jdbc.username}"*/>

<beans:propertyname=*"password"*value=*"${app.jdbc.password}"*/>

</beans:bean>

<beans:beanid=*"sqlMapClient1"*class=*"org.springframework.orm.ibatis.SqlMapClientFactoryBean"*>

<beans:propertyname=*"configLocation"*value=*"classpath:sql-maps-config.xml"*/>

<beans:propertyname=*"dataSource"*ref=*"dataSource1"*/>

</beans:bean>

<beans:beanid=*"transactionManager1"*

class=*"org.springframework.jdbc.datasource.DataSourceTransactionManager"*>

<beans:propertyname=*"dataSource"*ref=*"dataSource1"*/>

</beans:bean>

</beans:beans>

**Jdbc.properties:**

app.jdbc.driverClassName=com.mysql.jdbc.Driver

app.jdbc.url=jdbc:mysql://localhost:3306/springdb

app.jdbc.username=root

app.jdbc.password=

**sql-maps-config.xml:**

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<!DOCTYPEsqlMapConfig

PUBLIC"-//ibatis.apache.org//DTD SQL Map Config 2.0//EN"

"http://ibatis.apache.org/dtd/sql-map-config-2.dtd">

<sqlMapConfig>

<settingsuseStatementNamespaces=*"true"*/>

<transactionManagertype=*"JDBC"*>

<dataSourcetype=*"SIMPLE"*>

<propertyname=*"JDBC.Driver"*value=*"com.mysql.jdbc.Driver"*/>

<propertyname=*"JDBC.ConnectionURL"* value=*"jdbc:mysql://localhost:3306/hibernate"*/>

<propertyname=*"JDBC.Username"*value=*"root"*/>

<propertyname=*"JDBC.Password"*value=*""*/>

</dataSource>

</transactionManager>

<sqlMapresource=*"user.xml"*/>

</sqlMapConfig>

**User.xml:**

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<!DOCTYPEsqlMapPUBLIC"-//ibatis.apache.org//DTD SQL Map 2.0//EN""http://ibatis.apache.org/dtd/sql-map-2.dtd">

<sqlMapnamespace=*"user"*>

<typeAliasalias=*"User"*type=*"com.beingjavaguys.domain.User"*/>

<resultMapid=*"userResultMap"*class=*"User"*>

<resultproperty=*"userId"*column=*"user\_id"*/>

<resultproperty=*"firstName"*column=*"first\_name"*/>

<resultproperty=*"lastName"*column=*"last\_name"*/>

<resultproperty=*"gender"*column=*"gender"*/>

<resultproperty=*"city"*column=*"city"*/>

</resultMap>

<selectid=*"listUsers"*resultMap=*"userResultMap"*>

SELECT \* FROM user

</select>

<insertid=*"insertUser"*parameterClass=*"User"*>

INSERT INTO USER

(FIRST\_NAME,LAST\_NAME,GENDER,CITY)

VALUES(#firstName#,#lastName#,#gender#,#city#);

</insert>

<deleteid=*"deleteUser"*parameterClass=*"java.lang.Long"*>

DELETE FROM USER WHERE USER\_ID = #value#

</delete>

<selectid=*"getUserById"*parameterClass=*"java.lang.Long"*

resultMap=*"userResultMap"*>

SELECT \* FROM USER WHERE USER\_ID = #value#

</select>

<updateid=*"updateUser"*parameterClass=*"User"*>

UPDATE USER

SET FIRST\_NAME = #firstName#,

LAST\_NAME = #lastName#,

GENDER = #gender#,

CITY = #city#

WHERE USER\_ID = #userId#

</update>

</sqlMap>

**Log4j.xml:**

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<!DOCTYPElog4j:configurationSYSTEM"log4j.dtd">

<log4j:configurationdebug=*"true"*

xmlns:log4j=*'http://jakarta.apache.org/log4j/'*>

<!-- Console -->

<appendername=*"console"*class=*"org.apache.log4j.ConsoleAppender"*>

<layoutclass=*"org.apache.log4j.PatternLayout"*>

<paramname=*"ConversionPattern"*

value=*"%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L - %m%n"*/>

</layout>

</appender>

<!-- file -->

<appendername=*"file"*class=*"org.apache.log4j.RollingFileAppender"*>

<paramname=*"append"*value=*"false"*/>

<paramname=*"maxFileSize"*value=*"10KB"*/>

<paramname=*"maxBackupIndex"*value=*"5"*/>

<paramname=*"file"*value=*"F:/Coding/SpringCoding/WAE/logs/logs.log"*/>

<layoutclass=*"org.apache.log4j.PatternLayout"*>

<paramname=*"ConversionPattern"*

value=*"%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L - %m%n"*/>

</layout>

</appender>

<root>

<levelvalue=*"DEBUG"*/>

<appender-refref=*"console"*/>

<appender-refref=*"file"*/>

</root>

</log4j:configuration>

**User.java:**

**package** com.beingjavaguys.domain;

**publicclass** User {

**privateint**userId;

**private** String firstName;

**private** String lastName;

**private** String gender;

**private** String city;

**publicint** getUserId() {

**return**userId;

}

**publicvoid** setUserId(**int** userId) {

**this**.userId = userId;

}

**public** String getFirstName() {

**return**firstName;

}

**publicvoid** setFirstName(String firstName) {

**this**.firstName = firstName;

}

**public** String getLastName() {

**return**lastName;

}

**publicvoid** setLastName(String lastName) {

**this**.lastName = lastName;

}

**public** String getGender() {

**return**gender;

}

**publicvoid** setGender(String gender) {

**this**.gender = gender;

}

**public** String getCity() {

**return**city;

}

**publicvoid** setCity(String city) {

**this**.city = city;

}

}

**HomePageController.java:**

**package** com.beingjavaguys.controller;

**import** java.util.ArrayList;

**import** java.util.HashMap;

**import** java.util.List;

**import** java.util.Locale;

**import** java.util.Map;

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

**import** org.springframework.stereotype.Controller;

**import** org.springframework.ui.Model;

**import** org.springframework.validation.BindingResult;

**import** org.springframework.web.bind.annotation.ModelAttribute;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestMethod;

**import** org.springframework.web.bind.annotation.RequestParam;

**import** org.springframework.web.servlet.ModelAndView;

**import** com.beingjavaguys.domain.User;

**import** com.beingjavaguys.services.UserService;

@Controller

**publicclass** HomePageController {

@Autowired

UserService userService;

@RequestMapping("/addregister")

**public** ModelAndView getRegisterForm(@ModelAttribute("user") User user,

BindingResult result) {

**returnnew** ModelAndView("register");

}

@RequestMapping("/register")

**public** ModelAndView registerUser(@ModelAttribute User user,

BindingResult result) {

List<String> genderList = **new** ArrayList<String>();

genderList.add("male");

genderList.add("female");

List<String> cityList = **new** ArrayList<String>();

cityList.add("delhi");

cityList.add("gurgaon");

cityList.add("meerut");

cityList.add("noida");

Map<String, List> map = **new** HashMap<String, List>();

map.put("genderList", genderList);

map.put("cityList", cityList);

**returnnew** ModelAndView("register", "map", map);

}

@RequestMapping("/insert")

**public** String inserData(@ModelAttribute User user) {

**if** (user != **null**)

userService.insertData(user);

**return**"redirect:/getList";

}

@RequestMapping("/getList")

**public** ModelAndView getUserLIst() {

List<User> userList = userService.getUserList();

**returnnew** ModelAndView("userList", "userList", userList);

}

@RequestMapping("/edit")

**public** ModelAndView editUser(@RequestParam Long id,

@ModelAttribute User user) {

user = userService.getUser(id);

List<String> genderList = **new** ArrayList<String>();

genderList.add("male");

genderList.add("female");

List<String> cityList = **new** ArrayList<String>();

cityList.add("delhi");

cityList.add("gurgaon");

cityList.add("meerut");

cityList.add("noida");

Map<String, Object> map = **new** HashMap<String, Object>();

map.put("genderList", genderList);

map.put("cityList", cityList);

map.put("user", user);

**returnnew** ModelAndView("edit", "map", map);

}

@RequestMapping("/update")

**public** String updateUser(@ModelAttribute User user) {

userService.updateData(user);

**return**"redirect:/getList";

}

@RequestMapping("/delete")

**public** String deleteUser(@RequestParam Long id) {

System.*out*.println("id = " + id);

userService.deleteData(id);

**return**"redirect:/getList";

}

@RequestMapping(value = "/", method = RequestMethod.*GET*)

**public** String home(Locale locale, Model model) {

**return**"home";

}

}

**UserService.java:**

**package** com.beingjavaguys.services;

**import** java.util.List;

**import** com.beingjavaguys.domain.User;

**publicinterface** UserService {

**publicvoid** insertData(User user);

**public** List<User> getUserList();

**publicvoid** deleteData(Long id);

**public** User getUser(Long id);

**publicvoid** updateData(User user);

}

**UserServiceImpl.java:**

**package** com.beingjavaguys.services;

**import** java.util.List;

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

**import** com.beingjavaguys.dao.UserDao;

**import** com.beingjavaguys.domain.User;

**publicclass** UserServiceImpl **implements** UserService {

@Autowired

UserDao userdao;

@Override

**publicvoid** insertData(User user) {

userdao.insertData(user);

}

@Override

**public** List<User> getUserList() {

**return**userdao.getUserList();

}

@Override

**publicvoid** deleteData(Long id) {

userdao.deleteData(id);

}

@Override

**public** User getUser(Long id) {

**return**userdao.getUser(id);

}

@Override

**publicvoid** updateData(User user) {

userdao.updateData(user);

}

}

**UserDao.java:**

**package** com.beingjavaguys.dao;

**import** java.util.List;

**import** com.beingjavaguys.domain.User;

**publicinterface** UserDao {

**public** Long insertData(User user);

**public** List<User> getUserList();

**publicvoid** updateData(User user);

**publicvoid** deleteData(Long id);

**public** User getUser(Long id);

}

**UserDaoImpl.java:**

**package** com.beingjavaguys.dao;

**import** java.util.List;

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

**import** org.springframework.beans.factory.annotation.Qualifier;

**import** org.springframework.orm.ibatis.support.SqlMapClientDaoSupport;

**import** org.springframework.stereotype.Repository;

**import** com.ibatis.sqlmap.client.SqlMapClient;

**import** com.beingjavaguys.domain.User;

**import** com.beingjavaguys.dao.UserDao;

@Repository

**publicclass** UserDaoImpl **extends** SqlMapClientDaoSupport **implements** UserDao {

**privatestaticfinal** String *NAMESPACE* = "user.";

@Autowired

@Qualifier("sqlMapClient1")

**publicvoid** injectSqlMapClient(SqlMapClient sqlMapClient) {

setSqlMapClient(sqlMapClient);

}

@Override

@SuppressWarnings("unchecked")

**public** List<User> getUserList() {

**return** getSqlMapClientTemplate().queryForList(*NAMESPACE* + "listUsers");

}

@Override

**public** Long insertData(User user) {

**return** (Long) getSqlMapClientTemplate().insert(

*NAMESPACE* + "insertUser", user);

}

@Override

**publicvoid** updateData(User user) {

getSqlMapClientTemplate().update(*NAMESPACE* + "updateUser", user);

}

@Override

**publicvoid** deleteData(Long id) {

getSqlMapClientTemplate().delete(*NAMESPACE* + "deleteUser", id);

}

@Override

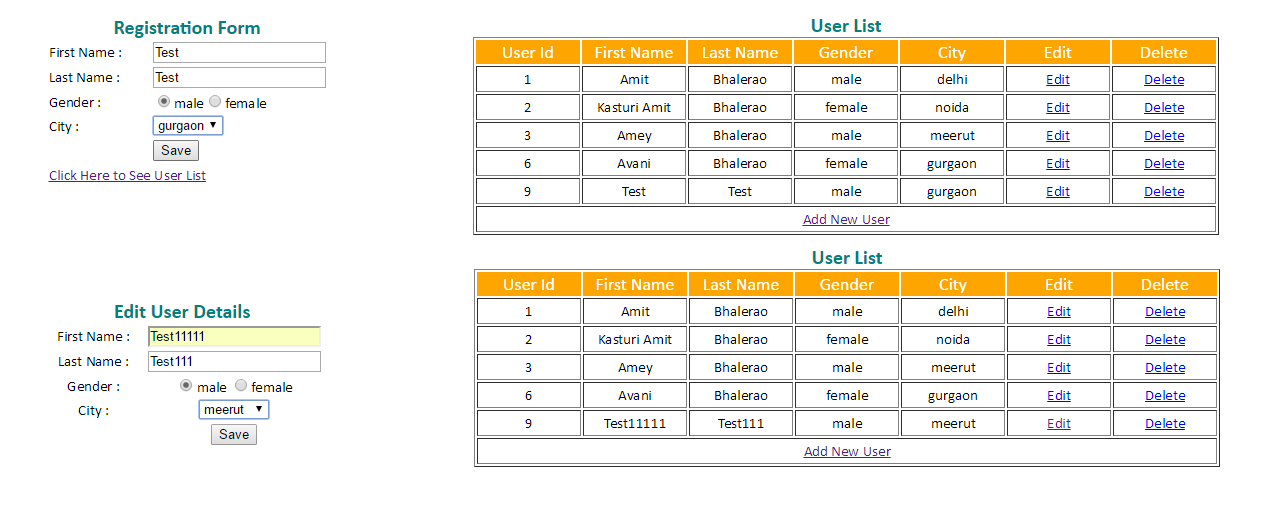
**public** User getUser(Long id) {

**return** (User) getSqlMapClientTemplate().queryForObject(

*NAMESPACE* + "getUserById", id);

}

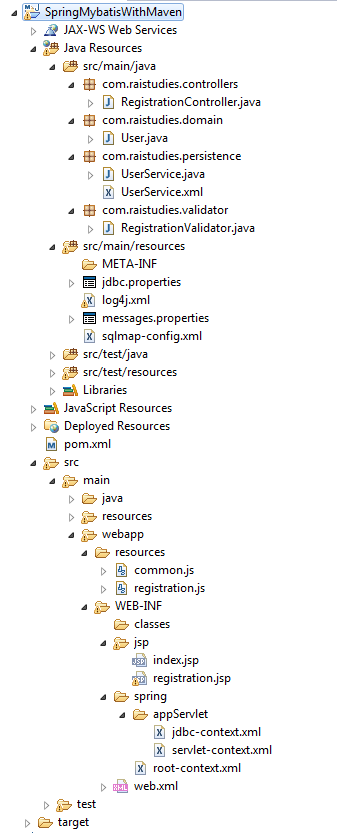
}

****

* Spring MyBatis Integration

1. **Using XML Mapping Configuration**

**Project Structure :**

****

**Pom.xml:**

<dependency>

<groupId>aopalliance</groupId>

<artifactId>aopalliance</artifactId>

<version>1.0</version>

</dependency>

<dependency>

<groupId>c3p0</groupId>

<artifactId>c3p0</artifactId>

<version>0.9.1.2</version>

</dependency>

<dependency>

<groupId>commons-logging</groupId>

<artifactId>commons-logging</artifactId>

<version>1.1.1</version>

</dependency>

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis-spring</artifactId>

<version>1.2.2</version>

</dependency>

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis</artifactId>

<version>3.2.7</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.1.21</version>

</dependency>

<!-- Servlet -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>servlet-api</artifactId>

<version>2.5</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>javax.servlet.jsp</groupId>

<artifactId>jsp-api</artifactId>

<version>2.1</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

<!-- AspectJ -->

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjrt</artifactId>

<version>${org.aspectj-version}</version>

</dependency>

<!-- @Inject -->

<dependency>

<groupId>javax.inject</groupId>

<artifactId>javax.inject</artifactId>

<version>1</version>

</dependency>

<!-- Spring -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-asm</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>org.springframework.beans</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${org.springframework-version}</version>

<exclusions>

<!-- Exclude Commons Logging in favor of SLF4j -->

<exclusion>

<groupId>commons-logging</groupId>

<artifactId>commons-logging</artifactId>

</exclusion>

</exclusions>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context-support</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<!-- Jdbc -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>org.springframework.expression</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<!-- AspectJ -->

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjrt</artifactId>

<version>${org.aspectj-version}</version>

</dependency>

<!-- Logging -->

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>${org.slf4j-version}</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>jcl-over-slf4j</artifactId>

<version>${org.slf4j-version}</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-log4j12</artifactId>

<version>${org.slf4j-version}</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>log4j</groupId>

<artifactId>log4j</artifactId>

<version>1.2.15</version>

<exclusions>

<exclusion>

<groupId>javax.mail</groupId>

<artifactId>mail</artifactId>

</exclusion>

<exclusion>

<groupId>javax.jms</groupId>

<artifactId>jms</artifactId>

</exclusion>

<exclusion>

<groupId>com.sun.jdmk</groupId>

<artifactId>jmxtools</artifactId>

</exclusion>

<exclusion>

<groupId>com.sun.jmx</groupId>

<artifactId>jmxri</artifactId>

</exclusion>

</exclusions>

<scope>runtime</scope>

</dependency>

<!-- @Inject -->

<dependency>

<groupId>javax.inject</groupId>

<artifactId>javax.inject</artifactId>

<version>1</version>

</dependency>

<!-- Test -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.7</version>

<scope>test</scope>

</dependency>

**Web.xml:**

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<web-appxmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns=*"http://java.sun.com/xml/ns/javaee"*

xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"*

version=*"2.5"*>

<context-param>

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

<param-value>/WEB-INF/spring/root-context.xml</param-value>

</context-param>

<listener>

<listener-class>

org.springframework.web.context.ContextLoaderListener

</listener-class>

</listener>

<servlet>

<servlet-name>appServlet</servlet-name>

<servlet-class>

org.springframework.web.servlet.DispatcherServlet

</servlet-class>

<init-param>

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

<param-value>

/WEB-INF/spring/appServlet/servlet-context.xml</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>appServlet</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

</web-app>

**Servlet-context.xml:**

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

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

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

xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:mvc=*"http://www.springframework.org/schema/mvc"*

xsi:schemaLocation=*"*

*http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*

*http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.0.xsd*

*http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd"*>

<mvc:resourcesmapping=*"/resources/\*\*"*location=*"/resources/"*/>

<!-- Application Message Bundle -->

<beanid=*"messageSource"*class=*"org.springframework.context.support.ReloadableResourceBundleMessageSource"*>

<propertyname=*"basename"*value=*"classpath:messages"*/>

<propertyname=*"cacheSeconds"*value=*"3000"*/>

</bean>

<!-- Scans the classpath of this application for @Components to deploy as beans -->

<context:component-scanbase-package=*"com.raistudies"*/>

<!-- Configures the @Controller programming model -->

<mvc:annotation-driven/>

<!-- Resolves view names to protected .jsp resources within the /WEB-INF/views directory -->

<beanid=*"viewResolver"*class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

<propertyname=*"prefix"*value=*"/WEB-INF/jsp/"*/>

<propertyname=*"suffix"*value=*".jsp"*/>

</bean>

<importresource=*"jdbc-context.xml"*/>

</beans>

**jdbc-context.xml:**

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

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

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*xmlns:p=*"http://www.springframework.org/schema/p"*

xmlns:tx=*"http://www.springframework.org/schema/tx"*xmlns:context=*"http://www.springframework.org/schema/context"*

xsi:schemaLocation=*"*

*http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*

*http://www.springframework.org/schema/tx*

*http://www.springframework.org/schema/tx/spring-tx-3.0.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context-3.0.xsd*

*"*>

<context:property-placeholderlocation=*"classpath:jdbc.properties"*/>

<!-- Enable annotation style of managing transactions -->

<tx:annotation-driventransaction-manager=*"transactionManager"*/>

<!-- Declare a datasource that has pooling capabilities -->

<beanid=*"dataSource"*class=*"com.mchange.v2.c3p0.ComboPooledDataSource"*

destroy-method=*"close"*p:driverClass=*"${app.jdbc.driverClassName}"*

p:jdbcUrl=*"${app.jdbc.url}"*p:user=*"${app.jdbc.username}"*p:password=*"${app.jdbc.password}"*

p:acquireIncrement=*"10"*p:idleConnectionTestPeriod=*"60"*p:maxPoolSize=*"100"*

p:maxStatements=*"50"*p:minPoolSize=*"10"*/>

<!-- Declare a transaction manager -->

<beanid=*"transactionManager"*

class=*"org.springframework.jdbc.datasource.DataSourceTransactionManager"*

p:dataSource-ref=*"dataSource"*/>

<!-- define the SqlSessionFactory, notice that configLocation is not needed when you use MapperFactoryBean -->

<beanid=*"sqlSessionFactory"*class=*"org.mybatis.spring.SqlSessionFactoryBean"*>

<propertyname=*"dataSource"*ref=*"dataSource"*/>

<propertyname=*"configLocation"*value=*"classpath:sqlmap-config.xml"*/>

</bean>

<!-- scan for mappers and let them be autowired -->

<beanclass=*"org.mybatis.spring.mapper.MapperScannerConfigurer"*>

<propertyname=*"basePackage"*value=*"com.raistudies.persistence"*/>

</bean>

</beans>

**Jdbc.properties:**

app.jdbc.driverClassName=com.mysql.jdbc.Driver

app.jdbc.url=jdbc:mysql://localhost/hibernate

app.jdbc.username=root

app.jdbc.password=

messages.properties:

field.name.empty=Namefieldismandatory.

field.standard.empty=Standardfieldismandatory.

field.age.empty=Agefieldismandatory.

field.sex.empty=Sexfieldismandatory.

field.age.NAN=Ageshouldbeanumber.

**sqlmap-config.xml:**

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<!DOCTYPEconfiguration

PUBLIC"-//mybatis.org//DTD Config 3.0//EN"

"http://mybatis.org/dtd/mybatis-3-config.dtd">

<configuration>

<settings>

<!-- changes from the defaults -->

<settingname=*"lazyLoadingEnabled"*value=*"false"*/>

</settings>

<typeAliases>

<typeAliastype=*"com.raistudies.domain.User"*alias=*"user"*/>

</typeAliases>

</configuration>

**UserService.xml:**

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<!DOCTYPEmapperPUBLIC"-//mybatis.org//DTD Mapper 3.0//EN"

"http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mappernamespace=*"com.raistudies.persistence.UserService"*>

<resultMapid=*"result"*type=*"user"*>

<resultproperty=*"id"*column=*"id"*/>

<resultproperty=*"name"*column=*"name"*/>

<resultproperty=*"standard"*column=*"standard"*/>

<resultproperty=*"age"*column=*"age"*/>

<resultproperty=*"sex"*column=*"sex"*/>

</resultMap>

<selectid=*"getAllUser"*resultMap=*"result"*>

SELECT id,name,standard,age,sex

FROM user;

</select>

<insertid=*"saveUser"*parameterType=*"user"*>

INSERT INTO user (name,standard,age,sex)

VALUE (#{name},#{standard},#{age},#{sex})

</insert>

<updateid=*"updateUser"*parameterType=*"user"*>

UPDATE user

SET

name = #{name},

standard = #{standard},

age = #{age},

sex = #{sex}

where id = #{id}

</update>

<deleteid=*"deleteUser"*parameterType=*"string"*>

DELETE FROM user

WHERE id = #{id}

</delete>

</mapper>

**User.java:**

**package** com.raistudies.domain;

**import** java.io.Serializable;

**publicclass** User **implements** Serializable {

**privatestaticfinallong***serialVersionUID* = 3647233284813657927L;

**private** String id;

**private** String name = **null**;

**private** String standard = **null**;

**private** String age;

**private** String sex = **null**;

**public** String getId() {

**return**id;

}

**publicvoid** setId(String id) {

**this**.id = id;

}

**public** String getName() {

**return**name;

}

**publicvoid** setName(String name) {

**this**.name = name;

}

**public** String getStandard() {

**return**standard;

}

**publicvoid** setStandard(String standard) {

**this**.standard = standard;

}

**public** String getAge() {

**return**age;

}

**publicvoid** setAge(String age) {

**this**.age = age;

}

**public** String getSex() {

**return**sex;

}

**publicvoid** setSex(String sex) {

**this**.sex = sex;

}

@Override

**public** String toString() {

**return**"User [name=" + name + ", standard=" + standard + ", age=" + age

+ ", sex=" + sex + "]";

}

}

**RegistrationController.java:**

**package** com.raistudies.controllers;

**import** java.util.List;

**import** java.util.UUID;

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

**import** org.springframework.stereotype.Controller;

**import** org.springframework.ui.ModelMap;

**import** org.springframework.validation.BindingResult;

**import** org.springframework.web.bind.annotation.ModelAttribute;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestMethod;

**import** org.springframework.web.servlet.ModelAndView;

**import** com.raistudies.domain.User;

**import** com.raistudies.persistence.UserService;

**import** com.raistudies.validator.RegistrationValidator;

@Controller

@RequestMapping(value="/registration")

**publicclass** RegistrationController {

**private** RegistrationValidator validator = **null**;

**private** UserService userService = **null**;

@Autowired

**publicvoid** setUserService(UserService userService) {

**this**.userService = userService;

}

**public** RegistrationValidator getValidator() {

**return**validator;

}

@Autowired

**publicvoid** setValidator(RegistrationValidator validator) {

**this**.validator = validator;

}

@RequestMapping(method=RequestMethod.*GET*)

**public** String showForm(ModelMap model){

List<User> users = userService.getAllUser();

model.addAttribute("users", users);

User user = **new** User();

user.setId(UUID.*randomUUID*().toString());

model.addAttribute("user", user);

**return**"registration";

}

@RequestMapping(value="/add", method=RequestMethod.*POST*)

**public** ModelAndView add(@ModelAttribute(value="user") User user,BindingResult result){

validator.validate(user, result);

ModelAndView mv = **new** ModelAndView("registration");

**if**(!result.hasErrors()){

userService.saveUser(user);

user = **new** User();

user.setId(UUID.*randomUUID*().toString());

mv.addObject("user", user);

}

mv.addObject("users", userService.getAllUser());

**return** mv;

}

@RequestMapping(value="/update", method=RequestMethod.*POST*)

**public** ModelAndView update(@ModelAttribute(value="user") User user,BindingResult result){

validator.validate(user, result);

ModelAndView mv = **new** ModelAndView("registration");

**if**(!result.hasErrors()){

userService.updateUser(user);

user = **new** User();

user.setId(UUID.*randomUUID*().toString());

mv.addObject("user", user);

}

mv.addObject("users", userService.getAllUser());

**return** mv;

}

@RequestMapping(value="/delete", method=RequestMethod.*POST*)

**public** ModelAndView delete(@ModelAttribute(value="user") User user,BindingResult result){

validator.validate(user, result);

ModelAndView mv = **new** ModelAndView("registration");

**if**(!result.hasErrors()){

userService.deleteUser(user.getId());

user = **new** User();

user.setId(UUID.*randomUUID*().toString());

mv.addObject("user", user);

}

mv.addObject("users", userService.getAllUser());

**return** mv;

}

}

**RegistrationValidator.java:**

**package** com.raistudies.validator;

**import** org.springframework.stereotype.Component;

**import** org.springframework.validation.Errors;

**import** org.springframework.validation.ValidationUtils;

**import** org.springframework.validation.Validator;

**import** com.raistudies.domain.User;

@Component

**publicclass**RegistrationValidator**implements** Validator {

**publicboolean** supports(Class<?> c) {

**return** User.**class**.isAssignableFrom(c);

}

**publicvoid** validate(Object command, Errors errors) {

ValidationUtils.*rejectIfEmptyOrWhitespace*(errors, "name",

"field.name.empty");

ValidationUtils.*rejectIfEmptyOrWhitespace*(errors, "standard",

"field.standard.empty");

ValidationUtils.*rejectIfEmptyOrWhitespace*(errors, "age",

"field.age.empty");

ValidationUtils.*rejectIfEmptyOrWhitespace*(errors, "sex",

"field.sex.empty");

User usrBean = (User) command;

**if** (!isNumber(usrBean.getAge().trim()))

errors.rejectValue("age", "field.age.NAN");

}

**privateboolean** isNumber(String str) {

**for** (**int** i = 0; i < str.length(); i++) {

// If we find a non-digit character we return false.

**if** (!Character.*isDigit*(str.charAt(i)))

**returnfalse**;

}

**returntrue**;

}

}

**UserService.java:**

**package** com.raistudies.persistence;

**import** java.util.List;

**import** com.raistudies.domain.User;

**publicinterface** UserService {

**publicvoid** saveUser(User user);

**publicvoid** updateUser(User user);

**publicvoid** deleteUser(String id);

**public** List<User> getAllUser();

}

**Registration.jsp:**

<!DOCTYPEHTMLPUBLIC"-//W3C//DTD XHTML 1.0 Transitional//EN""http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<%@pagesession=*"true"*%>

<%@tagliburi=*"http://www.springframework.org/tags/form"*prefix=*"form"*%>

<%@tagliburi=*"http://java.sun.com/jsp/jstl/core"*prefix=*"c"*%>

<htmlxmlns=*"http://www.w3.org/1999/xhtml"*>

<head>

<title>Spring MyBatis Integration</title>

<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">

<script type="text/javascript" src="<c:url value='/resources/common.js'/>"></script>

<script type="text/javascript" src="<c:url value='/resources/registration.js'/>"></script>

<script type="text/javascript">

**var** projectUrl = '<c:url value="/"/>';

**if**(projectUrl.indexOf(";", 0) != -1){

projectUrl = projectUrl.substring(0, projectUrl.indexOf(";", 0));

}

</script>

</head>

<body>

<fieldset>

<legend>Registration Form</legend>

<center>

<form:formcommandName=*"user"*action=*"/SpringMybatisWithMaven/registration/add"*name=*"userForm"*>

<form:hiddenpath=*"id"*/>

<table>

<tr><tdcolspan=*"2"*align=*"left"*><form:errorspath=*"\*"*cssStyle=*"color : red;"*/></td></tr>

<tr><td>Name : </td><td><form:inputpath=*"name"*/></td></tr>

<tr><td>Standard : </td><td><form:inputpath=*"standard"*/></td></tr>

<tr><td>Age : </td><td><form:inputpath=*"age"*/></td></tr>

<tr><td>Sex : </td><td><form:selectpath=*"sex"*>

<form:optionvalue=*"Male"*/>

<form:optionvalue=*"Female"*/>

</form:select></td></tr>

<tr><tdcolspan=*"2"*><inputtype=*"submit"*value=*"Save Changes"*/>

&nbsp;<inputtype=*"reset"*name=*"newUser"*value=*"New User"*onclick="setAddForm();"disabled=*"disabled"*/>

&nbsp;<inputtype=*"submit"*name=*"deleteUser"*value=*"Delete User"*onclick="setDeleteForm();"disabled=*"disabled"*/></td></tr>

</table>

</form:form>

</center>

</fieldset>

<c:iftest=*"*${!**empty** users}*"*>

<br/>

<center>

<tablewidth=*"90%"*>

<trstyle="background-color: *gray*;">

<th>Name</th>

<th>Standard</th>

<th>Age</th>

<th>Sex</th>

</tr>

<c:forEachitems=*"*${users}*"*var=*"user"*>

<trstyle="background-color: *silver*;" id=*"*${user.id}*"*onclick=*"setUpdateForm('*${user.id}*');"*>

<td><c:outvalue=*"*${user.name}*"*/></td>

<td><c:outvalue=*"*${user.standard}*"*/></td>

<td><c:outvalue=*"*${user.age}*"*/></td>

<td><c:outvalue=*"*${user.sex}*"*/></td>

</tr>

</c:forEach>

</table>

</center>

<br/>

</c:if>

</body>

</html>

**Common.js:**

function changeSelectValue(selectNode, value){

for(var index=0;index<selectNode.length;index++) {

if(selectNode[index].value == value)

selectNode.selectedIndex = index;

}

}

**Registration.js:**

var newUserId;

var addUrl = "/registration/add";

var deleteUrl = "/registration/delete";

var updateUrl = "/registration/update";

function setUpdateForm(element){

newUserId = document.userForm.id.value;

trElement = document.getElementById(element);

document.userForm.id.value = element;

document.userForm.name.value = trElement.children[0].innerHTML;

document.userForm.standard.value = trElement.children[1].innerHTML;

document.userForm.age.value = trElement.children[2].innerHTML;

changeSelectValue(document.userForm.sex, trElement.children[3].innerHTML);

document.userForm.newUser.disabled = false;

document.userForm.deleteUser.disabled = false;

document.userForm.action = projectUrl + updateUrl;

}

function setAddForm(){

document.userForm.id.value = newUserId;

document.userForm.name.value = "";

document.userForm.standard.value = "";

document.userForm.age.value = "";

changeSelectValue(document.userForm.sex, "Male");

document.userForm.newUser.disabled = true;

document.userForm.deleteUser.disabled = true;

document.userForm.action = projectUrl + addUrl;

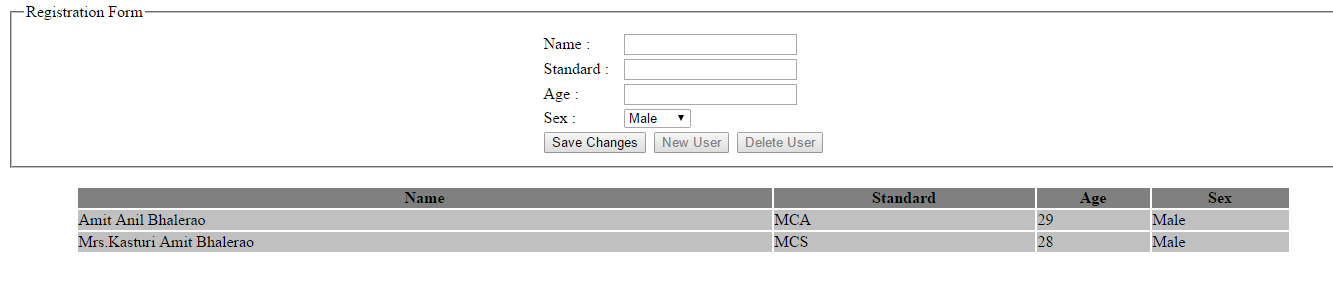
}

function setDeleteForm() {

document.userForm.action = projectUrl + deleteUrl;

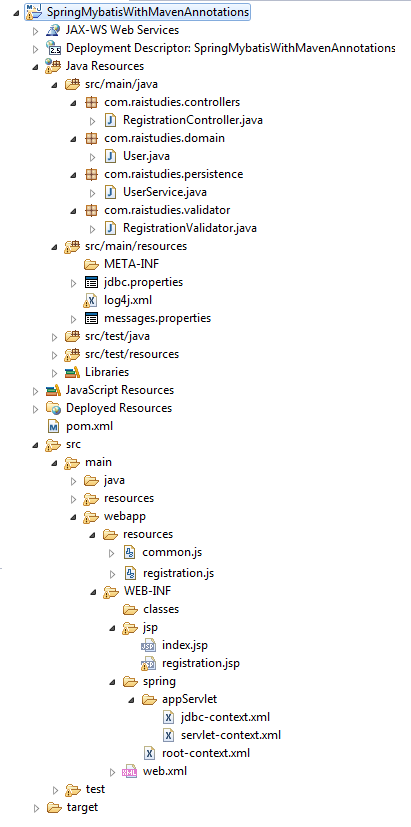
return true;

}



1. **Using Annotations**

**Project Structure:**



**Pom.xml:**

<dependency>

<groupId>aopalliance</groupId>

<artifactId>aopalliance</artifactId>

<version>1.0</version>

</dependency>

<dependency>

<groupId>c3p0</groupId>

<artifactId>c3p0</artifactId>

<version>0.9.1.2</version>

</dependency>

<dependency>

<groupId>commons-logging</groupId>

<artifactId>commons-logging</artifactId>

<version>1.1.1</version>

</dependency>

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis-spring</artifactId>

<version>1.2.2</version>

</dependency>

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis</artifactId>

<version>3.2.7</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.1.21</version>

</dependency>

<!-- Servlet -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>servlet-api</artifactId>

<version>2.5</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>javax.servlet.jsp</groupId>

<artifactId>jsp-api</artifactId>

<version>2.1</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

<!-- AspectJ -->

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjrt</artifactId>

<version>${org.aspectj-version}</version>

</dependency>

<!-- @Inject -->

<dependency>

<groupId>javax.inject</groupId>

<artifactId>javax.inject</artifactId>

<version>1</version>

</dependency>

<!-- Spring -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-asm</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>org.springframework.beans</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${org.springframework-version}</version>

<exclusions>

<!-- Exclude Commons Logging in favor of SLF4j -->

<exclusion>

<groupId>commons-logging</groupId>

<artifactId>commons-logging</artifactId>

</exclusion>

</exclusions>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context-support</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<!-- Jdbc -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>org.springframework.expression</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<!-- AspectJ -->

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjrt</artifactId>

<version>${org.aspectj-version}</version>

</dependency>

<!-- Logging -->

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>${org.slf4j-version}</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>jcl-over-slf4j</artifactId>

<version>${org.slf4j-version}</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-log4j12</artifactId>

<version>${org.slf4j-version}</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>log4j</groupId>

<artifactId>log4j</artifactId>

<version>1.2.15</version>

<exclusions>

<exclusion>

<groupId>javax.mail</groupId>

<artifactId>mail</artifactId>

</exclusion>

<exclusion>

<groupId>javax.jms</groupId>

<artifactId>jms</artifactId>

</exclusion>

<exclusion>

<groupId>com.sun.jdmk</groupId>

<artifactId>jmxtools</artifactId>

</exclusion>

<exclusion>

<groupId>com.sun.jmx</groupId>

<artifactId>jmxri</artifactId>

</exclusion>

</exclusions>

<scope>runtime</scope>

</dependency>

<!-- @Inject -->

<dependency>

<groupId>javax.inject</groupId>

<artifactId>javax.inject</artifactId>

<version>1</version>

</dependency>

<!-- Test -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.7</version>

<scope>test</scope>

</dependency>

**Web.xml:**

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<web-appxmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns=*"http://java.sun.com/xml/ns/javaee"*

xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"*

version=*"2.5"*>

<context-param>

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

<param-value>/WEB-INF/spring/root-context.xml</param-value>

</context-param>

<listener>

<listener-class>

org.springframework.web.context.ContextLoaderListener

</listener-class>

</listener>

<servlet>

<servlet-name>appServlet</servlet-name>

<servlet-class>

org.springframework.web.servlet.DispatcherServlet

</servlet-class>

<init-param>

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

<param-value>

/WEB-INF/spring/appServlet/servlet-context.xml

</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>appServlet</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

</web-app>

**Servlet-context.xml:**

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

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

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

xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:mvc=*"http://www.springframework.org/schema/mvc"*

xsi:schemaLocation=*"*

*http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*

*http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.0.xsd*

*http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd"*>

<mvc:resourcesmapping=*"/resources/\*\*"*location=*"/resources/"*/>

<!-- Application Message Bundle -->

<beanid=*"messageSource"*class=*"org.springframework.context.support.ReloadableResourceBundleMessageSource"*>

<propertyname=*"basename"*value=*"classpath:messages"*/>

<propertyname=*"cacheSeconds"*value=*"3000"*/>

</bean>

<!-- Scans the classpath of this application for @Components to deploy as beans -->

<context:component-scanbase-package=*"com.raistudies"*/>

<!-- Configures the @Controller programming model -->

<mvc:annotation-driven/>

<!-- Resolves view names to protected .jsp resources within the /WEB-INF/views directory -->

<beanid=*"viewResolver"*class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

<propertyname=*"prefix"*value=*"/WEB-INF/jsp/"*/>

<propertyname=*"suffix"*value=*".jsp"*/>

</bean>

<importresource=*"jdbc-context.xml"*/>

</beans>

**Jdbc-context.xml:**

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

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

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*xmlns:p=*"http://www.springframework.org/schema/p"*

xmlns:tx=*"*[*ttp://www.springframework.org/schema/tx"*](ttp://www.springframework.org/schema/tx%22)

xmlns:context=*"http://www.springframework.org/schema/context"*

xsi:schemaLocation=*"*

*http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*

*http://www.springframework.org/schema/tx*

*http://www.springframework.org/schema/tx/spring-tx-3.0.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context-3.0.xsd*

*"*>

<context:property-placeholderlocation=*"classpath:jdbc.properties"*/>

<!-- Enable annotation style of managing transactions -->

<tx:annotation-driventransaction-manager=*"transactionManager"*/>

<!-- Declare a datasource that has pooling capabilities -->

<beanid=*"dataSource"*class=*"com.mchange.v2.c3p0.ComboPooledDataSource"*

destroy-method=*"close"*p:driverClass=*"${app.jdbc.driverClassName}"*

p:jdbcUrl=*"${app.jdbc.url}"*p:user=*"${app.jdbc.username}"*p:password=*"${app.jdbc.password}"*

p:acquireIncrement=*"10"*p:idleConnectionTestPeriod=*"60"*p:maxPoolSize=*"100"*

p:maxStatements=*"50"*p:minPoolSize=*"10"*/>

<!-- Declare a transaction manager -->

<beanid=*"transactionManager"*

class=*"org.springframework.jdbc.datasource.DataSourceTransactionManager"*

p:dataSource-ref=*"dataSource"*/>

<!-- define the SqlSessionFactory, notice that configLocation is not needed when you use MapperFactoryBean -->

<beanid=*"sqlSessionFactory"*class=*"org.mybatis.spring.SqlSessionFactoryBean"*>

<propertyname=*"dataSource"*ref=*"dataSource"*/>

</bean>

<!-- scan for mappers and let them be autowired -->

<beanclass=*"org.mybatis.spring.mapper.MapperScannerConfigurer"*>

<propertyname=*"basePackage"*value=*"com.raistudies.persistence"*/>

</bean>

</beans>

**Jdbc.properties:**

app.jdbc.driverClassName=com.mysql.jdbc.Driver

app.jdbc.url=jdbc:mysql://localhost/hibernate

app.jdbc.username=root

app.jdbc.password=

**messages.properties:**

field.name.empty=Namefieldismandatory.

field.standard.empty=Standardfieldismandatory.

field.age.empty=Agefieldismandatory.

field.sex.empty=Sexfieldismandatory.

field.age.NAN=Ageshouldbeanumber.

**User.java:**

**package** com.raistudies.domain;

**import** java.io.Serializable;

**publicclass** User **implements** Serializable {

**privatestaticfinallong***serialVersionUID* = 3647233284813657927L;

**private** String id;

**private** String name = **null**;

**private** String standard = **null**;

**private** String age;

**private** String sex = **null**;

**public** String getId() {

**return**id;

}

**publicvoid** setId(String id) {

**this**.id = id;

}

**public** String getName() {

**return**name;

}

**publicvoid** setName(String name) {

**this**.name = name;

}

**public** String getStandard() {

**return**standard;

}

**publicvoid** setStandard(String standard) {

**this**.standard = standard;

}

**public** String getAge() {

**return**age;

}

**publicvoid** setAge(String age) {

**this**.age = age;

}

**public** String getSex() {

**return**sex;

}

**publicvoid** setSex(String sex) {

**this**.sex = sex;

}

@Override

**public** String toString() {

**return**"User [name=" + name + ", standard=" + standard + ", age=" + age

+ ", sex=" + sex + "]";

}

}

**RegistrationController.java:**

**package** com.raistudies.controllers;

**import** java.text.DateFormat;

**import** java.util.Date;

**import** java.util.List;

**import** java.util.Locale;

**import** java.util.UUID;

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

**import** org.springframework.stereotype.Controller;

**import** org.springframework.ui.Model;

**import** org.springframework.ui.ModelMap;

**import** org.springframework.validation.BindingResult;

**import** org.springframework.web.bind.annotation.ModelAttribute;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestMethod;

**import** org.springframework.web.servlet.ModelAndView;

**import** com.raistudies.domain.User;

**import** com.raistudies.persistence.UserService;

**import** com.raistudies.validator.RegistrationValidator;

@Controller

@RequestMapping(value="/registration")

**publicclass** RegistrationController {

**private** RegistrationValidator validator = **null**;

**private** UserService userService = **null**;

@Autowired

**publicvoid** setUserService(UserService userService) {

**this**.userService = userService;

}

**public** RegistrationValidator getValidator() {

**return**validator;

}

@Autowired

**publicvoid** setValidator(RegistrationValidator validator) {

**this**.validator = validator;

}

@RequestMapping(value = "/", method = RequestMethod.*GET*)

**public** String home(Locale locale, Model model) {

Date date = **new** Date();

DateFormat dateFormat = DateFormat.*getDateTimeInstance*(DateFormat.*LONG*, DateFormat.*LONG*, locale);

String formattedDate = dateFormat.format(date);

model.addAttribute("serverTime", formattedDate );

**return**"/registration";

}

@RequestMapping(method=RequestMethod.*GET*)

**public** String showForm(ModelMap model,@ModelAttribute(value="user") User user){

List<User> users = userService.getAllUser();

model.addAttribute("users", users);

user.setId(UUID.*randomUUID*().toString());

model.addAttribute("user", user);

**return**"registration";

}

@RequestMapping(value="/add", method=RequestMethod.*POST*)

**public** ModelAndView add(@ModelAttribute(value="user") User user,BindingResult result){

validator.validate(user, result);

ModelAndView mv = **new** ModelAndView("registration");

**if**(!result.hasErrors()){

userService.saveUser(user);

user = **new** User();

user.setId(UUID.*randomUUID*().toString());

mv.addObject("user", user);

}

mv.addObject("users", userService.getAllUser());

**return** mv;

}

@RequestMapping(value="/update", method=RequestMethod.*POST*)

**public** ModelAndView update(@ModelAttribute(value="user") User user,BindingResult result){

validator.validate(user, result);

ModelAndView mv = **new** ModelAndView("registration");

**if**(!result.hasErrors()){

userService.updateUser(user);

user = **new** User();

user.setId(UUID.*randomUUID*().toString());

mv.addObject("user", user);

}

mv.addObject("users", userService.getAllUser());

**return** mv;

}

@RequestMapping(value="/delete", method=RequestMethod.*POST*)

**public** ModelAndView delete(@ModelAttribute(value="user") User user,BindingResult result){

validator.validate(user, result);

ModelAndView mv = **new** ModelAndView("registration");

**if**(!result.hasErrors()){

userService.deleteUser(user.getId());

user = **new** User();

user.setId(UUID.*randomUUID*().toString());

mv.addObject("user", user);

}

mv.addObject("users", userService.getAllUser());

**return** mv;

}

}

**RegistrationValidator.java:**

**package** com.raistudies.validator;

**import** org.springframework.stereotype.Component;

**import** org.springframework.validation.Errors;

**import** org.springframework.validation.ValidationUtils;

**import** org.springframework.validation.Validator;

**import** com.raistudies.domain.User;

@Component

**publicclass**RegistrationValidator**implements** Validator {

**publicboolean** supports(Class<?> c) {

**return** User.**class**.isAssignableFrom(c);

}

**publicvoid** validate(Object command, Errors errors) {

ValidationUtils.*rejectIfEmptyOrWhitespace*(errors, "name", "field.name.empty");

ValidationUtils.*rejectIfEmptyOrWhitespace*(errors, "standard", "field.standard.empty");

ValidationUtils.*rejectIfEmptyOrWhitespace*(errors, "age", "field.age.empty");

ValidationUtils.*rejectIfEmptyOrWhitespace*(errors, "sex", "field.sex.empty");

User usrBean = (User)command;

**if**(!isNumber(usrBean.getAge().trim()))

errors.rejectValue("age", "field.age.NAN");

}

**privateboolean** isNumber(String str){

**for** (**int** i = 0; i < str.length(); i++) {

//If we find a non-digit character we return false.

**if** (!Character.*isDigit*(str.charAt(i)))

**returnfalse**;

}

**returntrue**;

}

}

**UserService.java:**

**package** com.raistudies.persistence;

**import** java.util.List;

**import** com.raistudies.domain.User;

**import** org.apache.ibatis.annotations.Insert;

**import** org.apache.ibatis.annotations.Options;

**import** org.apache.ibatis.annotations.Select;

**import** org.apache.ibatis.annotations.Update;

**import** org.apache.ibatis.annotations.Delete;

**publicinterface** UserService {

@Insert("INSERT INTO user (name,standard,age,sex) VALUES"

+ "(#{name},#{standard},#{age},#{sex})")

@Options(useGeneratedKeys = **true**, keyProperty = "id", flushCache = **true**, keyColumn = "id")

**publicvoid** saveUser(User user);

@Update("UPDATE user SET name = #{name},"

+ "standard = #{standard}," + "age = #{age}, " + "sex = #{sex} "

+ "where id = #{id}")

**publicvoid** updateUser(User user);

@Delete("DELETE FROM user WHERE id = #{id}")

**publicvoid** deleteUser(String id);

@Select("SELECT id,name,standard,age,sex FROM user")

**public** List<User> getAllUser();

}