Webflow for registration

1.Inside the pom.xml of backendproject add dependency.

a.Add the property inside properties:

<springwebflow.version>2.4.2.RELEASE</springwebflow.version>

b.Inside the dependencies , add :

<!-- Spring Web Flow dependencies -->

<dependency>

<groupId>org.springframework.webflow</groupId>

<artifactId>spring-webflow</artifactId>

<version>${springwebflow.version}</version>

</dependency>

2.Configure the dispatcher-servlet.xml of fronrendproject.

a.Add the following inside the beans parent tag:

xmlns:webflow=<http://www.springframework.org/schema/webflow-config>

b.Add the following inside the schemaLocation :

http://www.springframework.org/schema/webflow-config

<http://www.springframework.org/schema/webflow-config/spring-webflow-config.xsd>">

c.Add the following configurations and bean inside the parent beans tag as content,so file is:

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

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

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

xmlns:webflow=*"http://www.springframework.org/schema/webflow-config"*

xsi:schemaLocation=*"*

*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*

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

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

*http://www.springframework.org/schema/webflow-config*

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

<context:component-scan base-package=*"net.sports.ZenSportsFrontEnd"* />

<context:component-scan base-package=*"net.sports.ZenSportsBackEnd"* />

<!--Loading static resources -->

<mvc:annotation-driven />

<mvc:resources location=*"/assets/"* mapping=*"/resources/\*\*"* />

<bean id=*"multipartResolver"*

class=*"org.springframework.web.multipart.commons.CommonsMultipartResolver"* />

<!-- View Resolver -->

<bean id=*"viewResolver"*

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

<property name=*"prefix"* value=*"/WEB-INF/views/"* />

<property name=*"suffix"* value=*".jsp"* />

</bean>

<!-- WebFlow Configuration -->

<webflow:flow-executor id=*"flowExecutor"* />

<!-- is responsible for finding the flow file -->

<webflow:flow-registry id=*"flowRegistry"*

flow-builder-services=*"flowBuilderServices"* base-path=*"/WEB-INF"*>

<!-- here the id is mapped to invoke this flow -->

<webflow:flow-location id=*"register"* path=*"/memberShipFlow.xml"* />

</webflow:flow-registry>

<bean id=*"validator"*

class=*"org.springframework.validation.beanvalidation.LocalValidatorFactoryBean"* />

<!-- Reference the above bean below -->

<webflow:flow-builder-services id=*"flowBuilderServices"*

view-factory-creator=*"viewFactoryCreator"* validator=*"validator"* />

<!-- Webflow beans -->

<bean id=*"viewFactoryCreator"*

class=*"org.springframework.webflow.mvc.builder.MvcViewFactoryCreator"*>

<property name=*"viewResolvers"*>

<list>

<ref bean=*"viewResolver"* />

</list>

</property>

</bean>

<bean class=*"org.springframework.webflow.mvc.servlet.FlowHandlerAdapter"*>

<property name=*"flowExecutor"* ref=*"flowExecutor"* />

</bean>

<bean class=*"org.springframework.webflow.mvc.servlet.FlowHandlerMapping"*>

<property name=*"flowRegistry"* ref=*"flowRegistry"* />

<property name=*"order"* value=*"0"* />

</bean>

</beans>

3.Update the href in navbar of shared folder , :

<li id=*"register"*><a href=*"register"*><span

class=*"glyphicon glyphicon-plus"*></span> Register</a></li>

4.In the same level of dispatcher servlet.xml create another xml file , with the name memberShipFlow.xml .

5.Inside the controllers folder of the frontendproject , add a class file , with the name RegisterHandler,insdie it ,add:

@Component

**public** **class** RegisterHandler {

@Autowired

**private** IUserDAO userDAO;

**private** RegisterModel registerModel;

**public** RegisterModel initializeModel() {

**return** **new** RegisterModel();

}

// public User initializeUser()

// {

// return new User();

// }

**public** String validateUserDetails(User user) {

String status = "success";

**if** (user.getUserName().isEmpty()) {

status = "failure";

}

**if** (user.getUserFullName().isEmpty()) {

status = "failure";

}

**if** (user.getUserEmail().isEmpty()) {

status = "failure";

}

**if** (user.getUserMobile().isEmpty()) {

status = "failure";

}

**if** (user.getUserSecurityA().isEmpty()) {

status = "failure";

}

**return** status;

}

**public** String validateAddressDetails(Address address) {

String status = "success";

**if**(address.getAddAddress1().isEmpty())

{

status="failure";

}

**if**(address.getAddAddress2().isEmpty())

{

status="failure";

}

**if**(address.getAddCity().isEmpty())

{

status="failure";

}

**if**(address.getAddCountry().isEmpty())

{

status="failure";

}

**if**(address.getAddState().isEmpty())

{

status="failure";

}

**if**(address.getAddZipCode().isEmpty())

{

status="failure";

}

**return** status;

}}

6.Inside the model folder of the backendproject , add a class file , RegisterModel is the name.Inside it , add:

**public** **class** RegisterModel **implements** Serializable{

**public** **static** **long** getSerialversionuid() {

**return** ***serialVersionUID***;

}

/\*\*

\*

\*/

**private** **static** **final** **long** ***serialVersionUID*** = 1119460130208908635L;

User user;

Address shipping;

Address billing;

**public** RegisterModel() {

**super**();

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

**this**.shipping = **new** Address();

**this**.billing = **new** Address();

}

**public** User getUser() {

**return** user;

}

**public** **void** setUser(User user) {

**this**.user = user;

}

**public** Address getShipping() {

**return** shipping;

}

**public** **void** setShipping(Address shipping) {

**this**.shipping = shipping;

}

**public** Address getBilling() {

**return** billing;

}

**public** **void** setBilling(Address billing) {

**this**.billing = billing;

}}

7.Similar to serial id generation for this class , do the same for address , user,product and category class.Also generate getter and setter for the serial id in all the classes.

8.