Cart WebFlow

1. Create an xml file named cartFlow.xml inside the view folder of front end project and add the following code:

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

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

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

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

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

<var name="billing" class="com.health.HealthMedicineQuestBackEnd.model.Address"/>

<var name="shipping" class="com.health.HealthMedicineQuestBackEnd.model.Address"/>

<var name="cartitem" class="com.health.Health.HealthMedicineQuestBackEnd.model.CartItem"/>

<!-- Executed when webFlow is triggered -->

<on-start>

<!-- registerHandler is java class available through annotated RegisterHandler.java class. -->

<evaluate expression="cartHandler.initializeFlow()" result="flowScope.cartModel"/>

</on-start>

<view-state id="start" view="cart/cartview" model="cartitem">

<transition on="submit" to="billingAddress"/>

<on-exit>

<evaluate expression="cartHandler.saveCart(cartModel,cartitem)"></evaluate>

</on-exit>

</view-state>

<!-- User enters Billing Address -->

<view-state id="billingAddress" view="cart/billing" model="billing">

<transition on="submit" to="shippingAddress"/>

<on-exit>

<evaluate expression="cartHandler.saveBillingAddress(cartModel,billing)"></evaluate>

</on-exit>

</view-state>

<!-- User enters Billing Address -->

<view-state id="shippingAddress" view="cart/shipping" model="shipping">

<transition on="submit" to="orderpayment"/>

<on-exit>

<evaluate expression="registerHandler.saveShippingAddress(cartModel,shipping)"></evaluate>

</on-exit>

</view-state>

<!-- User view order receipt -->

<view-state id="orderpayment" view="cart/payment" model="pay">

<transition on="submit" to="orderreceipt"/>

<on-exit>

<evaluate expression="registerHandler.savePayment(cartModel,pay)"></evaluate>

</on-exit>

</view-state>

<!-- User view order receipt -->

<view-state id="orderreceipt" view="cart/orderreceipt" model="order">

<transition on="submit" to="confirm"/>

<on-exit>

<evaluate expression="registerHandler.saveOrderReceipt(cartModel,order)"></evaluate>

</on-exit>

</view-state>

<view-state id="confirm" view="cart/preOrder" model="flowScope.cartModel">

<transition on="edit" to="start" />

<transition on="submit" to="storedetail" />

</view-state>

<action-state id="storedetail">

<evaluate expression="cartHandler.storeDetail(cartModel)" ></evaluate>

<transition on="success" to="welcome" />

<transition on="failure" to="start" />

</action-state>

<end-state id="welcome" view="registration/welcome" />

<end-state id="toHome" view="externalRedirect:contextRelative:/home" />

<end-state id="toException" view="externalRedirect:contextRelative:exception.jsp" />

<!-- these events are available in every state-->

<global-transitions>

<transition on="home" to="toHome" />

<transition on="error" to="toException" />

</global-transitions>

</flow>

1. Create a class “CartModel” inside the model package of backend project. And add the following code:

**package** com.health.HealthMedicineQuestBackEnd.model;

**import** java.io.Serializable;

**public** **class** CartModel **implements** Serializable{

**private** **static** **final** **long** ***serialVersionUID*** = -4551827186947380158L;

CartItem cartItem;

Address billing;

Address shipping;

//Payment payment;

//Order order;

// public Payment getPayment() {

// return payment;

// }

//

// public void setPayment(Payment payment) {

// this.payment = payment;

// }

//

// public Order getOrder() {

// return order;

// }

//

// public void setOrder(Order order) {

// this.order = order;

// }

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

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

}

**public** CartItem getCartItem() {

**return** cartItem;

}

**public** **void** setCartItem(CartItem cartItem) {

**this**.cartItem = cartItem;

}

**public** Address getBilling() {

**return** billing;

}

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

**this**.billing = billing;

}

**public** Address getShipping() {

**return** shipping;

}

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

**this**.shipping = shipping;

}

**public** CartModel() {

cartItem=**new** CartItem();

billing=**new** Address();

shipping=**new** Address();

//order=new Order();

//payment=new Payment();

}

}

1. Open dispatcher-servlet.xml file and add following code after the line foe register:

<webflow:flow-location id="cart" path="/cartFlow.xml" />

1. Create a class inside the handler package of frontend project named CartHandler and add following code:

**package** com.health.HealthMedicineQuestFrontEnd.handler;

**import** com.health.HealthMedicineQuestBackEnd.model.Address;

**import** com.health.HealthMedicineQuestBackEnd.model.CartItem;

**import** com.health.HealthMedicineQuestBackEnd.model.CartModel;

**public** **class** CartHandler {

**public** CartModel initializeModel(){

**return** **new** CartModel();

}

**public** **void** saveCartItem(CartModel cartModel,CartItem cartItem){

cartModel.setCartItem(cartItem);

}

**public** **void** saveBillingAddress(CartModel cartModel,Address billing){

cartModel.setBillingAddress(billing);

}

**public** **void** saveShippingAddress(CartModel cartModel,Address shipping){

cartModel.setShippingAddress(shipping);

}

// public void saveOrder(CartModel cartModel,Order order){

// cartModel.setOrder(order);

// }

// public void savePayment(CartModel cartModel,Payment payment){

// cartModel.setPayment(payment);

// }

//

**public** **void** storeDetail(CartModel cartModel){

}

}