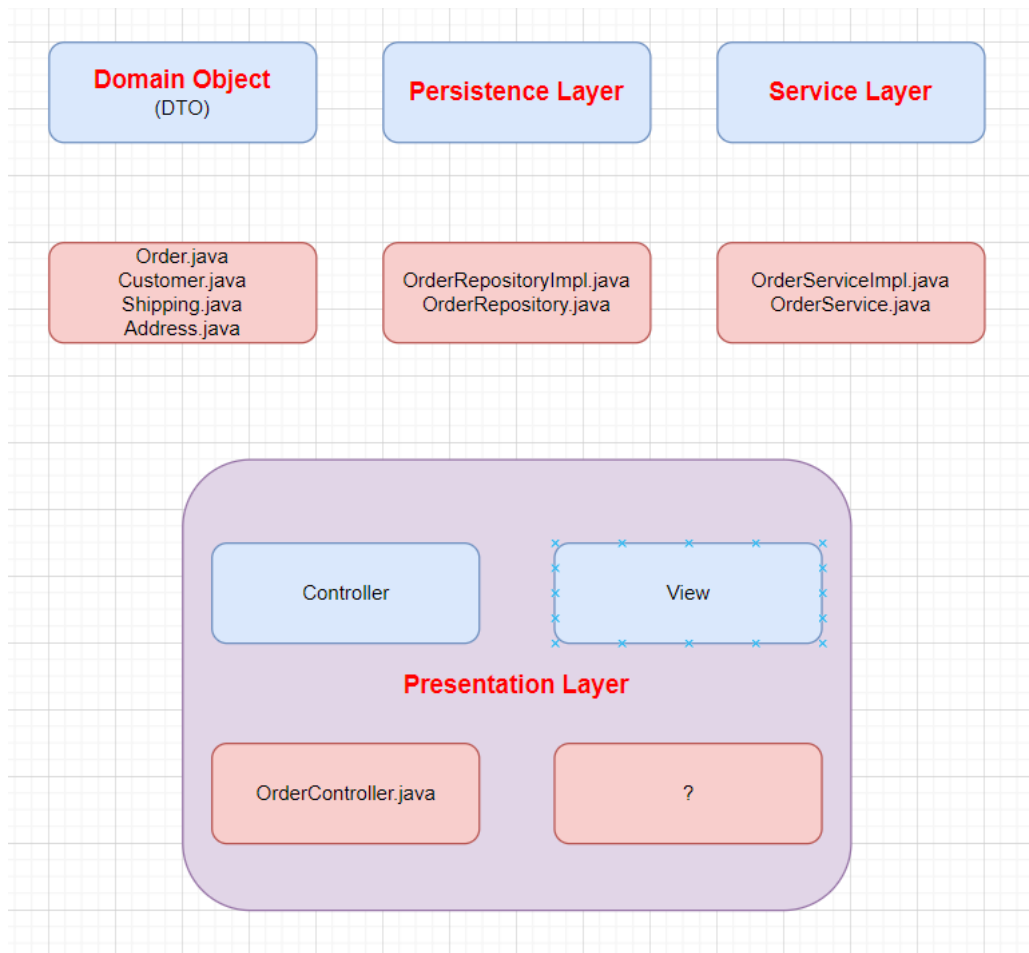


# Spring Day 8

OrderRepositoryImpl.java



관련 클래스들 모두 직렬화

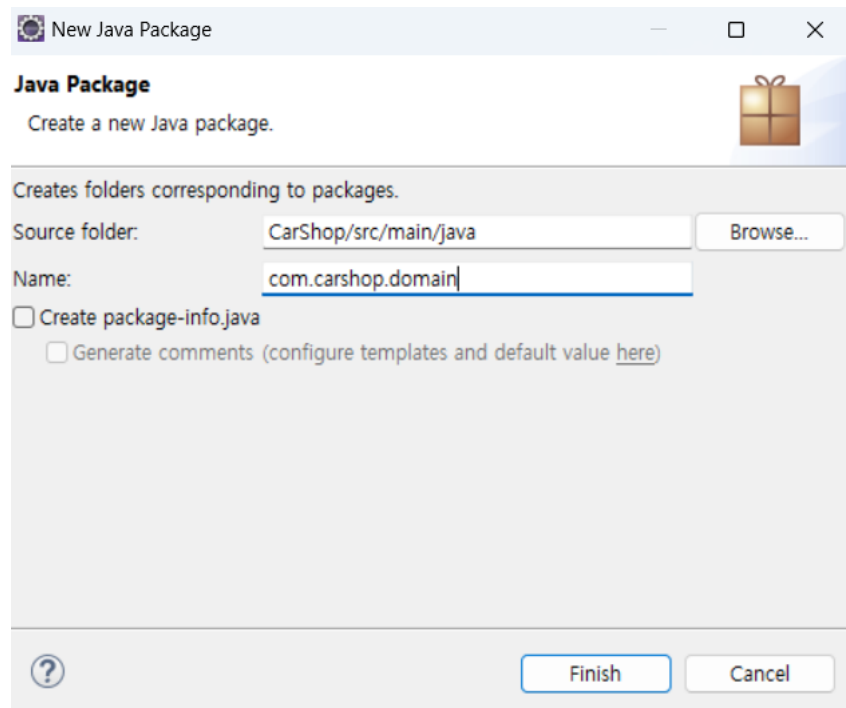
```
@SuppressWarnings("serial")
public class CartItem implements Serializable {
```

```
@SuppressWarnings("serial")
public class Cart implements Serializable {
```

```
@SuppressWarnings("serial")
public class CarDTO implements Serializable {
    //private static final long serialVersionUID = 6754974598729825L;
```

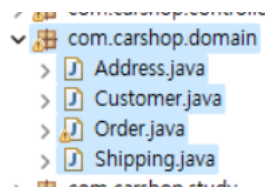
지금은 편의상 domain 패키지를 만들어 플로우에 관련된 모든 클래스들을 위치 시키지만

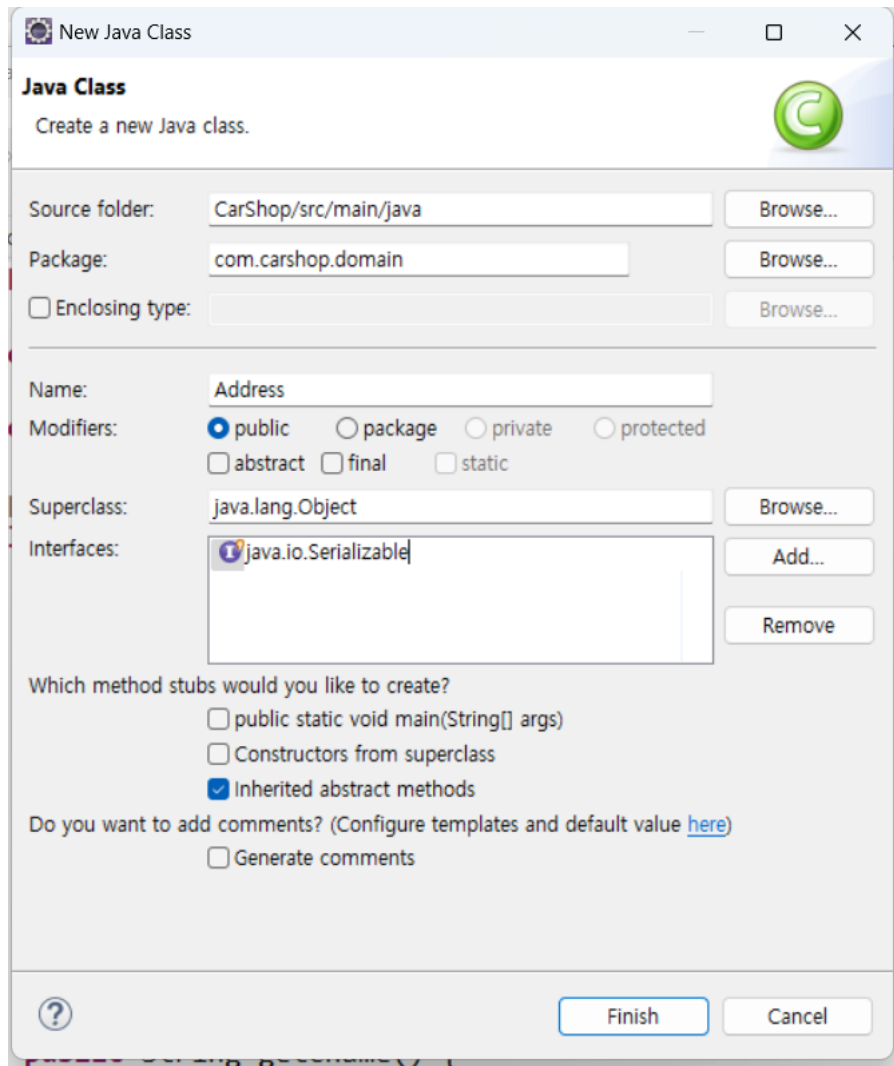
원래는 service 별, dto 별, repository별 로 각각 패키지를 만들어 사용하는 것을 권장한다.



주문처리와 관련된 새로운 클래스 4개 작성 , 전부 직렬화

Address.java Customer.java Shipping.java Order.java





```
package com.carshop.domain;

import java.io.Serializable;

@SuppressWarnings("serial")
public class Address implements Serializable {

    private String detailName; //세부주소
    private String addressName; //주소
    private String country; //국가명
    private String zipCode; //우편번호

    public String getDetailName() {
        return detailName;
    }
    public void setDetailName(String detailName) {
        this.detailName = detailName;
    }
    public String getAddressName() {
        return addressName;
    }
    public void setAddressName(String addressName) {
        this.addressName = addressName;
    }
    public String getCountry() {
        return country;
    }
    public void setCountry(String country) {
        this.country = country;
    }
    public String getZipCode() {
        return zipCode;
    }
}
```

```

public void setZipCode(String zipCode) {
    this.zipCode = zipCode;
}

@Override
public int hashCode() {
    final int prime = 31;
    int result = 1;
    result = prime * result + ((addressName == null) ? 0 : addressName.hashCode());
    result = prime * result + ((country == null) ? 0 : country.hashCode());
    result = prime * result + ((detailName == null) ? 0 : detailName.hashCode());
    result = prime * result + ((zipCode == null) ? 0 : zipCode.hashCode());
    return result;
}

@Override
public boolean equals(Object obj) {
    if (this == obj)
        return true;
    if (obj == null)
        return false;
    if (getClass() != obj.getClass())
        return false;
    Address other = (Address) obj;
    if (addressName == null) {
        if (other.addressName != null)
            return false;
    } else if (!addressName.equals(other.addressName))
        return false;
    if (country == null) {
        if (other.country != null)
            return false;
    } else if (!country.equals(other.country))
        return false;
    if (detailName == null) {
        if (other.detailName != null)
            return false;
    } else if (!detailName.equals(other.detailName))
        return false;
    if (zipCode == null) {
        if (other.zipCode != null)
            return false;
    } else if (!zipCode.equals(other.zipCode))
        return false;
    return true;
}
}

```

```

package com.carshop.domain;

import java.io.Serializable;

@SuppressWarnings("serial")
public class Customer implements Serializable {

    private String customerId; //고객id
    private String name; //고객성명
    private Address address; //고객주소 객체
    private String phone; //고객전화번호

    //기본생성자
    public Customer() {
        this.address = new Address();
    }

    public Customer(String customerId, String name) {
        this();
        this.customerId = customerId;
        this.name = name;
    }

    public String getCustomerId() {
        return customerId;
    }

    public void setCustomerId(String customerId) {
        this.customerId = customerId;
    }
}

```

```

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public Address getAddress() {
        return address;
    }

    public void setAddress(Address address) {
        this.address = address;
    }

    public String getPhone() {
        return phone;
    }

    public void setPhone(String phone) {
        this.phone = phone;
    }

    @Override
    public int hashCode() {
        final int prime = 31;
        int result = 1;
        result = prime * result + ((customerId == null) ? 0 : customerId.hashCode());
        return result;
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj)
            return true;
        if (obj == null)
            return false;
        if (getClass() != obj.getClass())
            return false;
        Customer other = (Customer) obj;
        if (customerId == null) {
            if (other.customerId != null)
                return false;
        } else if (!customerId.equals(other.customerId))
            return false;
        return true;
    }

}

```

```

package com.carshop.domain;

import java.io.Serializable;
import java.util.Date;

import org.springframework.format.annotation.DateTimeFormat;

@SuppressWarnings("serial")
public class Shipping implements Serializable {

    private String name; //배송 받는 고객 이름
    @DateTimeFormat(pattern="yyyy/MM/dd") //날짜 패턴을 제한한다. 오류 발생
    private Date date; //java.util.date
    private Address address; // 배송 주소 객체

    //기본 생성자
    public Shipping() {
        this.address = new Address();
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}

```

```

    }

    public Date getDate() {
        return date;
    }

    public void setDate(Date date) {
        this.date = date;
    }

    public Address getAddress() {
        return address;
    }

    public void setAddress(Address address) {
        this.address = address;
    }
}

```

```

package com.carshop.domain;

import java.io.Serializable;

import com.carshop.controller.Cart;

@SuppressWarnings("serial")
public class Order implements Serializable {

    private Long orderId; //주문 아이디
    private Cart cart; //장바구니 객체
    private Customer customer; //고객 객체
    private Shipping shipping; //배송지 객체

    //기본 생성자
    public Order() {
        this.customer = new Customer();
        this.shipping = new Shipping();
    }

    public Long getOrderId() {
        return orderId;
    }

    public void setOrderId(Long orderId) {
        this.orderId = orderId;
    }

    public Cart getCart() {
        return cart;
    }

    public void setCart(Cart cart) {
        this.cart = cart;
    }

    public Customer getCustomer() {
        return customer;
    }

    public void setCustomer(Customer customer) {
        this.customer = customer;
    }

    public Shipping getShipping() {
        return shipping;
    }

    public void setShipping(Shipping shipping) {
        this.shipping = shipping;
    }

    @Override
    public int hashCode() {
        final int prime = 31;
        int result = 1;
        result = prime * result + ((orderId == null) ? 0 : orderId.hashCode());
        return result;
    }
}

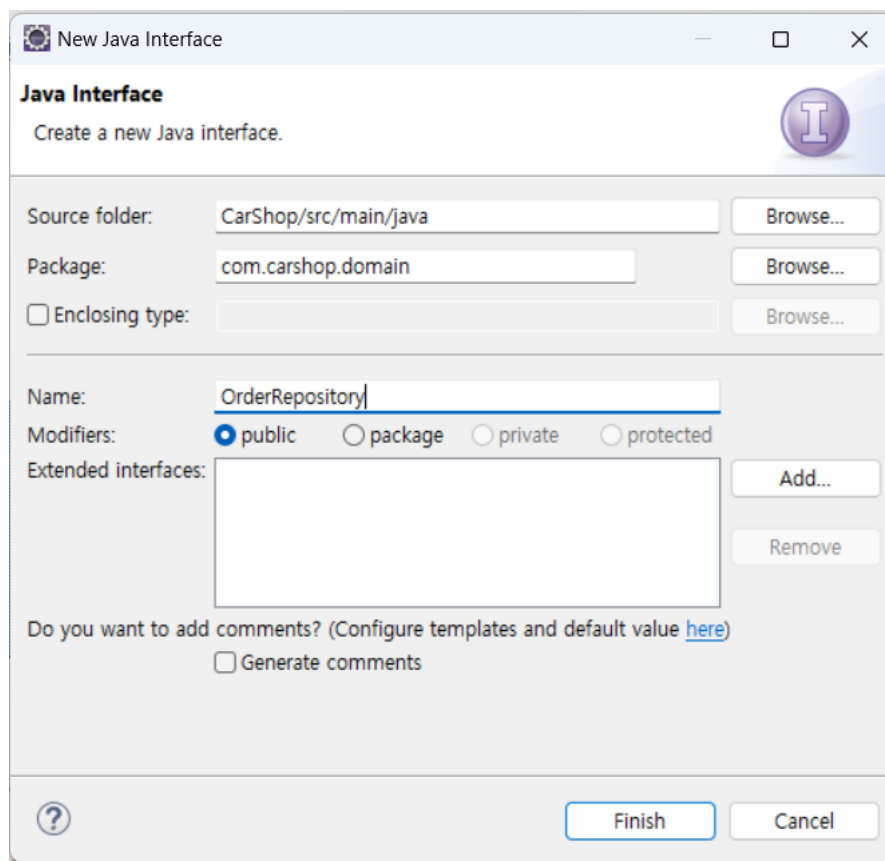
```

```

@Override
public boolean equals(Object obj) {
    if (this == obj)
        return true;
    if (obj == null)
        return false;
    if (getClass() != obj.getClass())
        return false;
    Order other = (Order) obj;
    if (orderId == null) {
        if (other.orderId != null)
            return false;
    } else if (!orderId.equals(other.orderId))
        return false;
    return true;
}
}

```

## Repository



## OrderRepository.java

```

package com.carshop.domain;

public interface OrderRepository {
    Long saveOrder(Order order);
}

```

## OrderRepositoryImpl.java

```
package com.carshop.domain;

import java.util.HashMap;
import java.util.Map;

import org.springframework.stereotype.Repository;

@Repository
public class OrderRepositoryImpl implements OrderRepository {

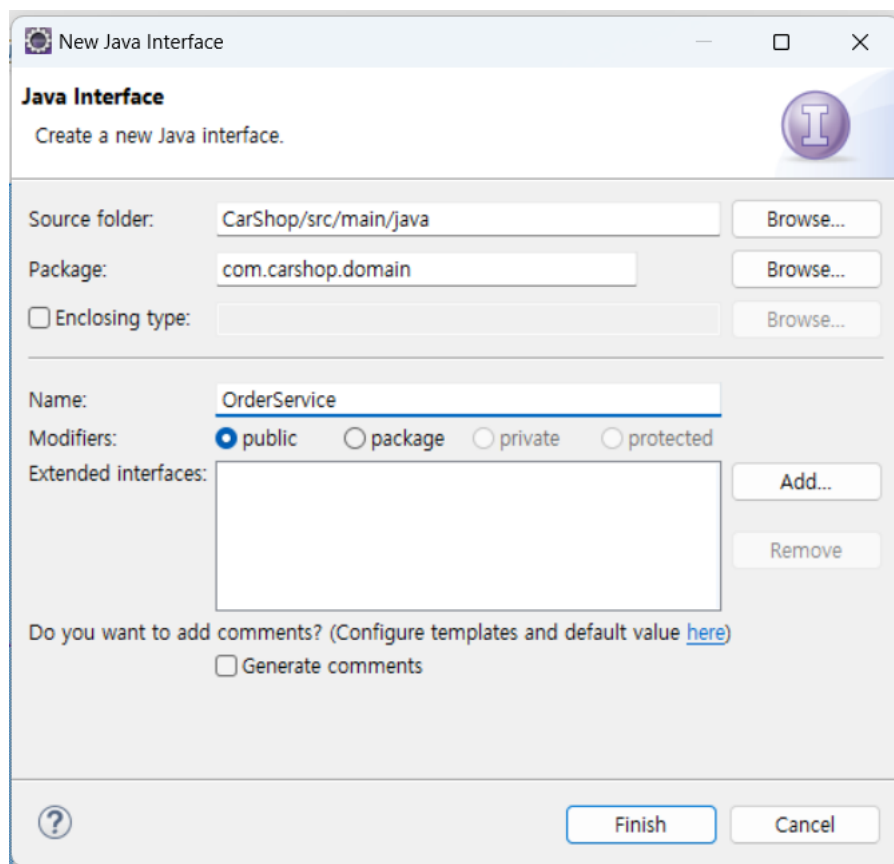
    private Map<Long, Order> listOfOrders;
    private long nextOrderId;

    public OrderRepositoryImpl() {
        listOfOrders = new HashMap<Long, Order>();
        nextOrderId = 2000;
    }

    @Override
    public Long saveOrder(Order order) {
        order.setOrderId(getNextOrderId());
        listOfOrders.put(order.getOrderId(), order);

        return order.getOrderId();
    }

    private synchronized long getNextOrderId() {
        return nextOrderId++;
    }
}
```



```
package com.carshop.domain;

public interface OrderService {
```



```

    void confirmOrder(String cid, long quantity);
    Long saveOrder(Order order);
}

```

```

package com.carshop.domain;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.carshop.controller.CartService;

@Service
public class OrderServiceImpl implements OrderService {

    @Autowired
    private OrderRepository orderRepository;

    @Autowired
    private CartService cartService;

    @Override
    public void confirmOrder(String cid, long quantity) {
        // 추후에는 판매가 이루어지면 재고량에서 1개를 빼주는 계산 처리
    }

    @Override
    public Long saveOrder(Order order) {
        Long orderId = orderRepository.saveOrder(order);
        cartService.delete(order.getCart().getCartId());
        return orderId;
        //주문이 처리가 되어 기존 장바구니는 비워준다
    }
}

```

#### OrderController.java

```

package com.carshop.domain;

import org.springframework.stereotype.Controller;

@Controller
public class OrderController {

    //Order 컨트롤러는 실제 작동하지는 않는다. webflow가 주문처리를 담당한다.
}

```

## webflow 환경 설정

1. pom.xml 의존성 라이브러리 추가



Maven
Gradle
Gradle (Short)
Gradle (Kotlin)
SBT
Ivy
Grape

Leiningen
Builder

```

<!--
https://mvnrepository.com/artifact/org.springframework.webflow/spring-
webflow -->
<dependency>
  <groupId>org.springframework.webflow</groupId>
  <artifactId>spring-webflow</artifactId>
  <version>2.5.1.RELEASE</version>

```

☒ Include comment with link to declaration

```

<!-- https://mvnrepository.com/artifact/org.springframework.webflow/spring-webflow -->
<dependency>
  <groupId>org.springframework.webflow</groupId>
  <artifactId>spring-webflow</artifactId>
  <version>2.5.1.RELEASE</version>
</dependency>

```

```

</dependency>
  <groupId>commons-fileupload</groupId>
  <artifactId>commons-fileupload</artifactId>
  <version>1.5</version>
</dependency>
<!-- https://mvnrepository.com/artifact/commons-io/commons-io -->
<dependency>
  <groupId>commons-io</groupId>
  <artifactId>commons-io</artifactId>
  <version>2.11.0</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.springframework.webflow/spring-webflow -->
<dependency>
  <groupId>org.springframework.webflow</groupId>
  <artifactId>spring-webflow</artifactId>
  <version>2.5.1.RELEASE</version>
</dependency>

```

## 2. servlet-context.xml 웹플로우 설정

```

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

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

```

```
<?xml version="1.0" encoding="UTF-8"?>
<beans:beans xmlns="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"
  xmlns:webflow="http://www.springframework.org/schema/webflow-config"
  xsi:schemaLocation="http://www.springframework.org/schema/mvc https://www.springframework.org/schema/
    http://www.springframework.org/schema/beans https://www.springframework.org/schema/beans/spring-b
    http://www.springframework.org/schema/webflow-config http://www.springframework.org/schema/webflo
    http://www.springframework.org/schema/context https://www.springframework.org/schema/context/spri
```

```
<webflow:flow-registry id="flowRegistry">
  <webflow:flow-location path="/WEB-INF/flows/order/order-flow.xml" id="order"/>
  <webflow:flow-location path="/WEB-INF/flows/ftest/ftest-flow.xml" id="ftest"/>
</webflow:flow-registry>

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

<beans:bean id="flowHandlerMapping"
  class="org.springframework.webflow.mvc.servlet.FlowHandlerMapping">
  <beans:property name="flowRegistry" ref="flowRegistry" />
</beans:bean>

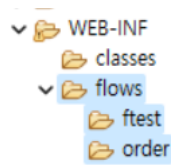
<beans:bean id="flowHandlerAdapter" class="org.springframework.webflow.mvc.servlet.FlowHandlerAdapter">
  <beans:property name="flowExecutor" ref="flowExecutor" />
</beans:bean>
```

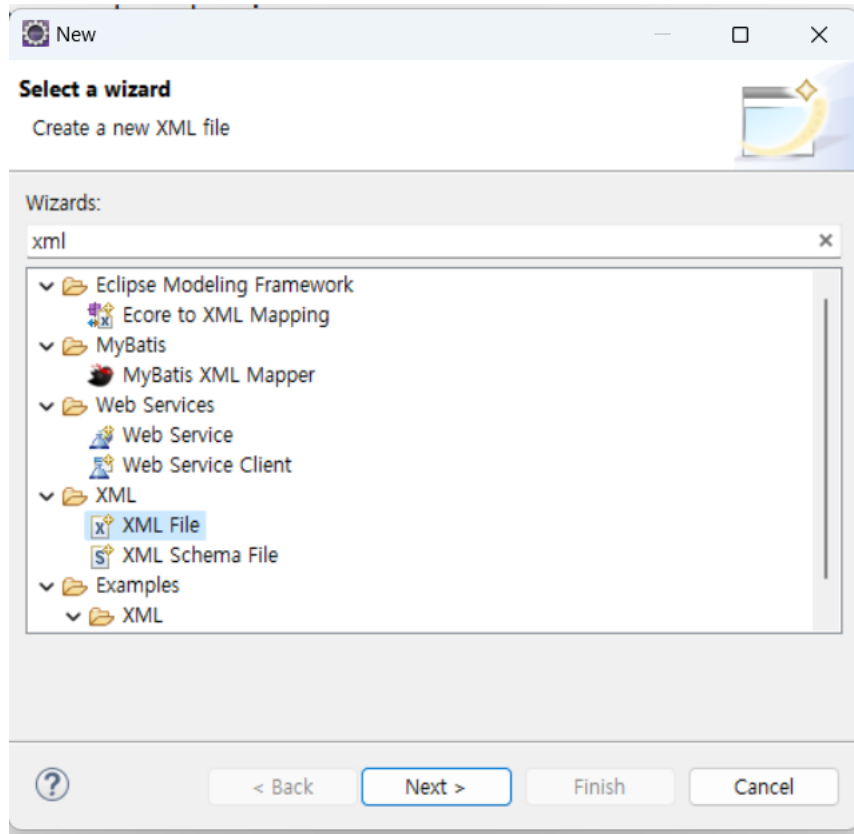
나머지 설정은 항상 그대로 넣어주고 아래 부분만 변경한다.

필요한 플로우를 여러개 설정하여 사용할수 있다. 위치와 아이디로 접근한다.

```
<webflow:flow-registry id="flowRegistry">
  <webflow:flow-location path="/WEB-INF/flows/order/order-flow.xml" id="order"/>
  <webflow:flow-location path="/WEB-INF/flows/ftest/ftest-flow.xml" id="ftest"/>
</webflow:flow-registry>
```

웹 플로우는 WEB-INF 바로 아래에 flows 아래에 ftest 연습용 order 실제적용 폴더를 생성하자





```
<?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="fetest" class="com.carshop.domain.Fetest" />

</flow>
```

```
<?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="ftest" class="com.carshop.domain.Ftest" />

</flow>
```

<var name="ftest" class="com.carshop.domain.Ftest" />

name 은 해당 플로우의 name값으로 호출이 되어 플로우가 진행된다.

class 매칭이되는 클래스를 명목상 하나 연결한다. (직렬화)

```
package com.carshop.domain;

import java.io.Serializable;

@SuppressWarnings("serial")
public class Ftest implements Serializable {

    // 명목상 지정하는 클래스
}
```

<flow> .....</flow> 모든 웹 플로는 <flow> 시작하여 </flow> 끝난다.

사이에 상태 state 또는 전환 transition 으로 정의한다.

```
<view-state id="ftest1">
    <transition to="ftest2" />
</view-state>
```

<view-state> 요소는 웹 플로우에서 보여줄 뷰 jsp 를 정의한다 id값은 뷰 jsp 페이지와 매핑된다.

```

<view-state id="fctest1">
  <transition to="fctest2" />
</view-state>

<view-state id="fctest2" view="fctest2.jsp">
  <transition to="fctest2" />
</view-state>

```

위에서 처럼 view 파일의 이름을 직접 지정하면 해당 이름의 뷰가 보여지고 생략하면 id값과 일치하는 뷰로 자동 선택 된다.

!!! 주의 항상 xml 설정파일과 같은 위치에 jsp 를 생성하고 만약 id 값에 해당하는 혹은 view 에 지정한 파일이 없을 경우에는 404 오류가 발생한다.

<transition> 요소는 전환하는 역할을 한다. 따라서 to=fctest2 를 지정하면 fctest2 id 가 호출된다.

<end-state> 요소 는 페이지 흐름의 종료를 정의한다.

<end-state id = "fctestCancelled">

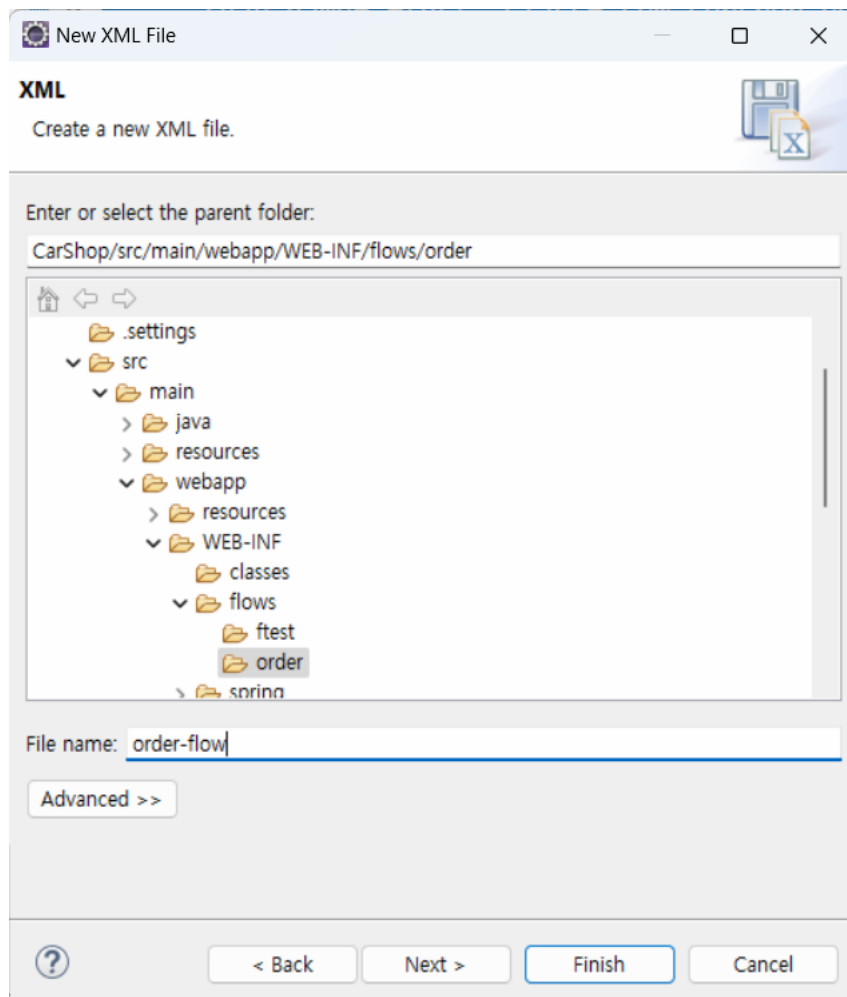
<action-state> 요소 특정 시점에 페이지 흐름을 제어 , 결과를 기반으로 다른 상태로 전환한다.

<evaluate> 매서드를 호출한 결과 값을 사용하여 분기

예) 주문내용을 디비에 저장하는 매서드를 호출 하였는데 실패할경우 혹은 성공할경우..

<global-transitions> ..... </global-transitions> 어느 흐름에서도 사용가능

## 프로젝트에 웹플로우 적용



```
<?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.xs

      <var name="order" class="com.carshop.domain.Order" />

</flow>
```

```
<?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="order" class="com.carshop.domain.Order" />
```



```
</flow>
```

```
<?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="order" class="com.carshop.domain.Order" />

    <view-state id="addCartToOrder">
        <transition to="orderCustomerInfo" />
    </view-state>

    <view-state id="orderCustomerInfo" model="order">
        <transition on="customerInfo" to="orderShippingInfo"/>
    </view-state>

    <view-state id="orderShippingInfo" model="order">
        <transition on="ShippingInfo" to="orderConfirmation"/>
        <transition on="backToCustomerInfo" to="orderCustomerInfo"/>
    </view-state>

    <view-state id="orderConfirmation">
        <transition on="orderConfirm" to="confirmOrder"/>
        <transition on="backToShippingInfo" to="orderShippingInfo"/>
    </view-state>

    <action-state id="confirmOrder">
        <evaluate expression="orderServiceImpl.saveOrder(order)" result="order.orderId"/>
        <transition to="orderFinished"/>
    </action-state>

    <view-state id="orderFinished" model="order">
        <transition to="endState"/>
    </view-state>

    <end-state id="endState"></end-state>

    <global-transitions>
        <transition on="cancel" to="endState"/>
    </global-transitions>

</flow>
```

## orderCustomerInfo.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>

    <link
        href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css"
        rel="stylesheet"
        integrity="sha384-GLhlTQ8iRABdZLL603oVMWsktQOp6b7In1ZL3/Jr59b6EGGoI1aFkw7cmDA6j6gD"
        crossorigin="anonymous">

    <nav class="navbar navbar-expand navbar-dark bg-dark fixed-top"
        aria-label="Second navbar example">
        <div class="container-fluid">
```

```

<a class="navbar-brand" href="#">CarShop</a>
<button class="navbar-toggler" type="button"
  data-bs-toggle="collapse" data-bs-target="#navbarsExample02"
  aria-controls="navbarsExample02" aria-expanded="false"
  aria-label="Toggle navigation">
  <span class="navbar-toggler-icon"></span>
</button>

<div class="collapse navbar-collapse" id="navbarsExample02">
  <ul class="navbar-nav me-auto">
    <li class="nav-item"><a class="nav-link active"
      aria-current="page" href="#">홈</a></li>
    <li class="nav-item"><a class="nav-link" href="/cars">차량
      보기</a></li>
    <li class="nav-item"><a class="nav-link" href="/boards">게시판</a>
    </li>
    <li class="nav-item dropdown"><a
      class="nav-link dropdown-toggle" href="#" role="button"
      data-bs-toggle="dropdown" aria-expanded="false">회원관리 </a>
    <ul class="dropdown-menu">
      <li><a class="dropdown-item" href="/login">로그인</a></li>
      <li><a class="dropdown-item" href="/logout">로그아웃</a></li>
    </ul></li>
  </ul>
  <c:choose>
    <c:when test="${isLogOn == true}">
      <form method="post" action="/logout">
        <input type="submit" class="btn btn-success" value="Logout" /> <input
          type="hidden" name="${_csrf.parameterName}"
          value="${_csrf.token}" />
      </form>
    </c:when>
    <c:otherwise>
      <form method="post" action="/login">
        <input type="submit" class="btn btn-primary" value="Login" /> <input
          type="hidden" name="${_csrf.parameterName}"
          value="${_csrf.token}" />
      </form>
    </c:otherwise>
  </c:choose>
</div>
</nav>
<div class="my-30">
  <div class="alert alert-dark">
    <div class="container">
      <h1>고객 정보</h1>
    </div>
  </div>
</div>

```

```

<div class="container">
  <form:form modelAttribute="order.customer" class="form-horizontal">
    <fieldset>
      <legend>고객 세부 사항</legend>
      <div class="form-group row">
        <label class="col-sm-2 control-label">고객 ID</label>
        <div class="col-sm-3">
          <form:input path="customerId" class="form-control" />
        </div>
      </div>
      <div class="form-group row">
        <label class="col-sm-2 control-label">성명</label>
        <div class="col-sm-3">
          <form:input path="name" class="form-control" />
        </div>
      </div>
      <div class="form-group row">
        <label class="col-sm-2 control-label">전화번호</label>
        <div class="col-sm-3">
          <form:input path="phone" class="form-control" />
        </div>
      </div>
      <div class="form-group row">
        <label class="col-sm-2 control-label">국가명</label>
        <div class="col-sm-3">
          <form:input path="address.country" class="form-control" />
        </div>
      </div>
      <div class="form-group row">
        <label class="col-sm-2 control-label">우편번호</label>
        <div class="col-sm-3">

```

```

        <form:input path="address.zipCode" class="form-control" />
    </div>
</div>
<div class="form-group row">
    <label class="col-sm-2 control-label">주소</label>
    <div class="col-sm-5">
        <form:input path="address.addressName" class="form-control" />
    </div>
</div>
<div class="form-group row">
    <label class="col-sm-2 control-label">세부주소</label>
    <div class="col-sm-3">
        <form:input path="address.detailName" class="form-control" />
    </div>
</div>
<input type="hidden" name="_flowExecutionKey"
    value="{flowExecutionKey}" />

<div class="form-group row">
    <div class="col-sm-offset-2 col-sm-10">
        <input type="submit" class="btn btn-primary" value="등록"
            name="_eventId_customerInfo" />
        <button class="btn btn-default" name="_eventId_cancel">취소</button>
    </div>
</div>

</fieldset>

</form:form>
</div>

```

```

<nav class="navbar fixed-bottom navbar-expand-sm navbar-dark bg-dark">
    <div class="container-fluid">
        <a class="navbar-brand" href="#">Car Shop</a>
        <button class="navbar-toggler" type="button"
            data-bs-toggle="collapse" data-bs-target="#navbarCollapse"
            aria-controls="navbarCollapse" aria-expanded="false"
            aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarCollapse">
            <ul class="navbar-nav">
                <li class="nav-item"><a class="nav-link active"
                    aria-current="page" href="/cars">자동차 보기</a></li>
            </ul>
        </div>
    </div>
</nav>

```

```

<script
    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-w76AqPfdKMBDXo30jS1Sgez6pr3x5MlQ1ZAGC+nuZB+EYdgRZgiwxhTBTkF7CXvN"
    crossorigin="anonymous"></script>

```

```

<script
    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-w76AqPfdKMBDXo30jS1Sgez6pr3x5MlQ1ZAGC+nuZB+EYdgRZgiwxhTBTkF7CXvN"
    crossorigin="anonymous"></script>

```

```

</body>
</html>

```

orderShippingInfo.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>

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

```

```

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

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>배송 세부 사항</title>
</head>
<body>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<link
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css"
    rel="stylesheet"
    integrity="sha384-GlhlTQ8iRABdZL603oVMWSktQOp6b7In1Zl3/Jr59b6EG6oI1aFkw7cmDA6j6gD"
    crossorigin="anonymous">

<nav class="navbar navbar-expand navbar-dark bg-dark fixed-top"
    aria-label="Second navbar example">
<div class="container-fluid">
    <a class="navbar-brand" href="#">CarShop</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
        data-bs-target="#navbarsExample02" aria-controls="navbarsExample02"
        aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse" id="navbarsExample02">
        <ul class="navbar-nav me-auto">
            <li class="nav-item"><a class="nav-link active"
                aria-current="page" href="#">홈</a></li>
            <li class="nav-item"><a class="nav-link" href="/cars">차량 보기</a></li>
            <li class="nav-item"><a class="nav-link" href="/boards">게시판</a>
            </li>
            <li class="nav-item dropdown"><a
                class="nav-link dropdown-toggle" href="#" role="button"
                data-bs-toggle="dropdown" aria-expanded="false"> 회원관리
            </a>
            <ul class="dropdown-menu">
                <li><a class="dropdown-item" href="/login">로그인</a></li>
                <li><a class="dropdown-item" href="/logout">로그아웃</a></li>
            </ul></li>
        </ul>
        <c:choose>
        <c:when test="${isLogOn == true}">
        <form method="post" action="/logout">
            <input type="submit" class="btn btn-success" value="Logout" />
            <input type="hidden" name="${_csrf.parameterName}" value="${_csrf.token}" />
        </form>
        </c:when>
        <c:otherwise>
        <form method="post" action="/login">
            <input type="submit" class="btn btn-primary" value="Login" />
            <input type="hidden" name="${_csrf.parameterName}" value="${_csrf.token}" />
        </form>
        </c:otherwise>
        </c:choose>
        </div>
    </div>
</nav>
<div class="my-5">
<div class="alert alert-dark">
    <div class="container">
        <h1>배송 세부 사항</h1>
    </div>
</div>
</div>

<div class="container">

    <form:form modelAttribute="order.shipping" class="form-horizontal">
    <fieldset>
    <legend>배송 세부 사항</legend>
    <div class="form-group row">
        <label class="col-sm-2 control-label">성명</label>
        <div class="col-sm-3">
            <form:input path="name" class="form-control" />
        </div>
    </div>
    <div class="form-group row">
        <label class="col-sm-2 control-label">배송일</label>
        <div class="col-sm-3">

```

```

        <form:input path="date" class="form-control" /> (yyyy/mm/dd)
    </div>
    <div class="col-sm-6">
        <form:errors path="date" cssClass="text-danger"/>
    </div>
</div>
<div class="form-group row">
    <label class="col-sm-2 control-label">국가명</label>
    <div class="col-sm-3">
        <form:input path="address.country" class="form-control" />
    </div>
</div>
<div class="form-group row">
    <label class="col-sm-2 control-label">우편번호</label>
    <div class="col-sm-3">
        <form:input path="address.zipCode" class="form-control" />
    </div>
</div>
<div class="form-group row">
    <label class="col-sm-2 control-label">주소</label>
    <div class="col-sm-5">
        <form:input path="address.addressName" class="form-control" />
    </div>
</div>
<div class="form-group row">
    <label class="col-sm-2 control-label">세부주소</label>
    <div class="col-sm-3">
        <form:input path="address.detailName" class="form-control" />
    </div>
</div>
<input type="hidden" name="_flowExecutionKey" value="{flowExecutionKey}"/>
<div class="form-group row">
    <div class="col-sm-offset-2 col-sm-10">
        <button class="btn btn-default" name="_eventId_backToCustomerInfo">이전</button>
        <input type="submit" class="btn btn-primary" value="등록" name="_eventId_shippingInfo"/>
        <button class="btn btn-default" name="_eventId_cancel">취소</button>
    </div>
</div>
</fieldset>
</form:form>
</div>

<nav class="navbar fixed-bottom navbar-expand-sm navbar-dark bg-dark">
    <div class="container-fluid">
        <a class="navbar-brand" href="#">Car Shop</a>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarCollapse" aria-controls="navbarCollapse"
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarCollapse">
            <ul class="navbar-nav">
                <li class="nav-item">
                    <a class="nav-link active" aria-current="page" href="/cars">자동차 보기</a>
                </li>
            </ul>
        </div>
    </div>
</nav>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js" integrity="sha384-w76AqPfdKMBDXo30jS"
    </script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js" integrity="sha384-w76AqPfdKMBDXo30jS"
    </script>
</body>
</html>

```

## orderConfirmation.jsp

```

<div class="container">
    <form:form modelAttribute="order" class="form-horizontal">
        <div class="well col-md-9 col-md-offset-2" style="background:#fafafe; padding: 20px">
            <div class="text-center">
                <h1>영수증</h1>
            </div>
        </div>
    </form:form>
</div>

```

```

</div>
<div class="row">
  <div class="col-md-6">
    <address>
      <strong>배송 주소</strong> <br>
      성명 : ${order.shipping.name}<br>
      우편번호 : ${order.shipping.address.zipCode}<br>
      주소 : ${order.shipping.address.addressName} ${order.shipping.address.detailName} (${order.shipping.address.country}) <br>
    </address>
  </div>
  <div class="col-md-6 text-right">
    <p> <em>배송일: <fmt:formatDate type="date" value="${order.shipping.date}" /></em></p>
  </div>
</div>
<div class="row">
  <div class="col-md-9">
    <address>
      <strong>청구주소</strong> <br>
      성명 : ${order.customer.name}<br>
      우편번호 : ${order.customer.address.zipCode}<br>
      주소 : ${order.customer.address.addressName} ${order.customer.address.detailName} (${order.customer.address.country})<br>
      HP : ${order.customer.phone}<br>
    </address>
  </div>
</div>
<div class="row">
  <table class="table table-hover">
    <thead>
      <tr><th>도서</th>
      <th>#</th>
      <th class="text-center">가격</th>
      <th class="text-center">소개</th>
    </tr>
    <tbody>
      <c:forEach var="cartItem" items="${order.cart.cartItems}">
        <tr>
          <td> <em>${cartItem.value.car.cname}</em></td>
          <td style="text-align: center">${cartItem.value.quantity}</td>
          <td class="text-center">${cartItem.value.car.cprice}</td>
          <td class="text-center">${cartItem.value.totalPrice}원</td>
        </tr>
      </c:forEach>
      <tr>
        <td> </td>
        <td> </td>
        <td class="text-right"><h5> <strong>총액: </strong></h5></td>
        <td class="text-center text-danger"><h4><strong>${order.cart.grandTotal}</strong></h4></td>
      </tr>
    </tbody>
  </table>
  <input type="hidden" name="_flowExecutionKey" value="${flowExecutionKey}" />
  <button class="btn btn-default" name="_eventId_backToShippingInfo">이전</button>
  <button type="submit" class="btn btn-success" name="_eventId_orderConfirmed"> 주문완료 </button>
  <button class="btn btn-default" name="_eventId_cancel">취소</button>
</div>
</div>
</form:form>
</div>

```

## orderFinished.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>주문 완료</title>
</head>
<body>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css"
rel="stylesheet"

```

```

integrity="sha384-GlhlTQ8iRABdZLl603oVMwSktQ0p6b7In1Zl3/Jr59b6EGGoI1aFkw7cmDA6j6gD"
crossorigin="anonymous">

<nav class="navbar navbar-expand navbar-dark bg-dark fixed-top"
aria-label="Second navbar example">
<div class="container-fluid">
<a class="navbar-brand" href="#">CarShop</a>
<button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarsExample02" aria-controls="navbarsExample02"
aria-expanded="false" aria-label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>

<div class="collapse navbar-collapse" id="navbarsExample02">
<ul class="navbar-nav me-auto">
<li class="nav-item"><a class="nav-link active"
aria-current="page" href="#">홈</a></li>
<li class="nav-item"><a class="nav-link" href="/cars">차량 보기</a></li>
<li class="nav-item"><a class="nav-link" href="/boards">게시판</a>
</li>
<li class="nav-item dropdown"><a
class="nav-link dropdown-toggle" href="#" role="button"
data-bs-toggle="dropdown" aria-expanded="false"> 회원관리
</a>
<ul class="dropdown-menu">
<li><a class="dropdown-item" href="/login">로그인</a></li>
<li><a class="dropdown-item" href="/logout">로그아웃</a></li>
</ul></li>
</ul>
<c:choose>
<c:when test="{isLogOn == true}">
<form method="post" action="/logout">
<input type="submit" class="btn btn-success" value="Logout" />
<input type="hidden" name="{_csrf.parameterName}" value="{_csrf.token}" />
</form>
</c:when>
<c:otherwise>
<form method="post" action="/login">
<input type="submit" class="btn btn-primary" value="Login" />
<input type="hidden" name="{_csrf.parameterName}" value="{_csrf.token}" />
</form>
</c:otherwise>
</c:choose>
</div>
</div>
</nav>
<div class="my-5">
<div class="alert alert-dark">
<div class="container">
<h1>주문 완료</h1>
</div>
</div>
</div>

<div class="container">
<h2 class="alert alert-danger">주문해 주셔서 감사합니다.</h2>
<p>주문 번호 : ${order.orderId}</p>
</div>

<div class="container">
<p><a href="/cars" class="btn btn-primary">제품목록</a></p>
</div>

<nav class="navbar fixed-bottom navbar-expand-sm navbar-dark bg-dark">
<div class="container-fluid">
<a class="navbar-brand" href="#">Car Shop</a>
<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarCollapse" aria-controls="navbarCollapse"
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarCollapse">
<ul class="navbar-nav">
<li class="nav-item">
<a class="nav-link active" aria-current="page" href="/cars">자동차 보기</a>
</li>
</ul>
</div>
</div>
</nav>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js" integrity="sha384-w76AqPfdkMBDXo30jS"
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js" integrity="sha384-w76AqPfdkMBDXo30jS"

```

```
</body>
</html>
```

예외 처리 3가지

CartServiceImpl.java

```
public Cart validateCart(String cartId) {
    Cart cart = cartRepository.read(cartId);
    if(cart == null || cart.getCartItems().size()==0) {
        throw new CartException(cartId);
    }
    return cart;
}
```

```
> Shipping.java
▼ com.carshop.exception
> CartException.java
> com.carshop.study
```

```
<action-state id = "addCartToOrder">
    <evaluate expression="cartServiceImpl.validateCart(requestParameters.
    <transition to="orderCartWarning" on-exception="com.carshop.exception
    <transition to="orderCustomerInfo"/>
</action-state>
```

```
<view-state id="orderCartWarning">
    <transition to="endState"/>
</view-state>
```

xml

```
<?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="order" class="com.carshop.domain.Order" />

    <action-state id = "addCartToOrder">
        <evaluate expression="cartServiceImpl.validateCart(requestParameters.cartId)" result="order.cart"/>
        <transition to="orderCartWarning" on-exception="com.carshop.exception.CartException" />
        <transition to="orderCustomerInfo"/>
    </action-state>

    <view-state id="orderCustomerInfo" model="order">
        <transition on="customerInfo" to="orderShippingInfo"/>
    </view-state>
```



```

<view-state id="orderShippingInfo" model="order">
  <transition on="shippingInfo" to="orderConfirmation"/>
  <transition on="backToCustomerInfo" to="orderCustomerInfo"/>
</view-state>

<view-state id="orderConfirmation">
  <transition on="orderConfirmed" to="confirmOrder"/>
  <transition on="backToShippingInfo" to="orderShippingInfo"/>
</view-state>

<action-state id="confirmOrder">
  <evaluate expression="orderServiceImpl.saveOrder(order)" result="order.orderId"/>
  <transition to="orderFinished"/>
</action-state>

<view-state id="orderFinished" model="order">
  <transition to="endState"/>
</view-state>

<view-state id="orderCartWarning">
  <transition to="endState"/>
</view-state>

<end-state id="endState"></end-state>

<global-transitions>
  <transition on="cancel" to="endState"/>
</global-transitions>

</flow>

```

## car.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form"%>

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Car Detail</title>
<script>
function addToCart(action) {
    document.addForm.action = action;
    document.addForm.submit();
    alert("제품이 장바구니에 추가되었습니다.")
    history.back();

    /*      window.location.reload(); */

}
</script>
</head>
<body>
<%@ include file="header.jsp" %>

<div class="alert alert-dark">
  <div class="container">
    <h1>차량 상세 보기</h1>
  </div>
</div>

<div class="container">
  <div class="row" align="center">
     " style="width:60%"/>

    <h3>${car.cid}</h3>
    <p>${car.cname}</p>
    <p>${car.cprice} 만원</p>
    <p>${car.ccate}</p>
    <p>${car.cdsc}</p>

```

```

    <p>
    <form:form name="addForm" method="put">
    <a href="javascript:addToCart('/cart/add/${car.cid}')" class="btn btn-primary">제품주문 &raquo;</a>
    <a href="c:url value="/cart"/>" class = "btn btn-warning">장바구니 &raquo;</a>
    <a href="c:url value="/cars"/>" class = "btn btn-secondary">제품목록 &raquo;</a>
    </form:form>
    </div>
</div>

<%@ include file="footer.jsp" %>
<br><br><br><br><br><br>
</body>
</html>

```