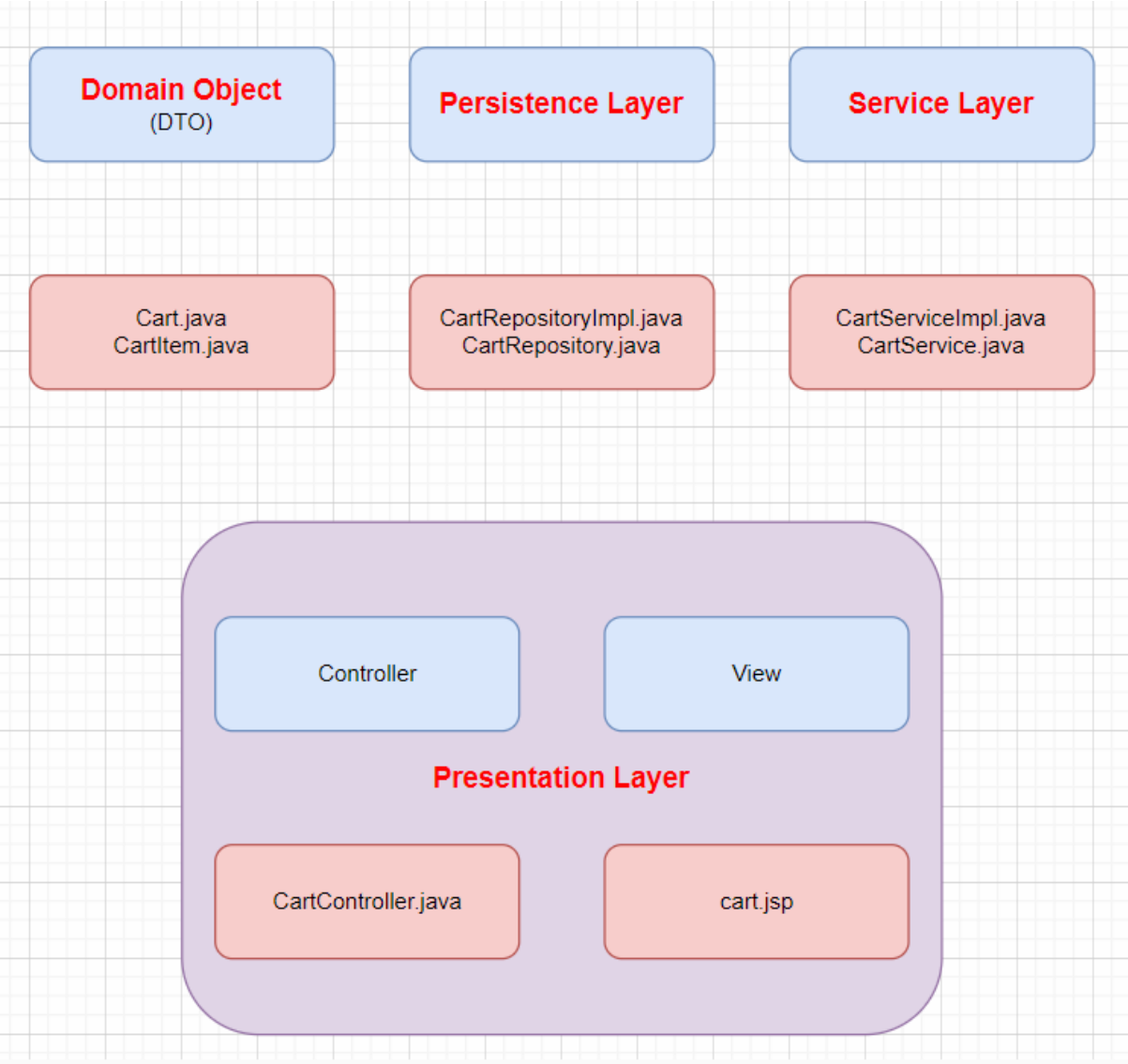
Spring Day 7 230308

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/ed2928a9-128f-48ac-940b-2121815c408c/CarShop7.zip



Cart

Service Layer

```
package com.carshop.controller;

public interface CartService {

  Cart create(Cart cart);
  Cart read(String cartId);
}
```

```
package com.carshop.controller;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class CartServiceImpl implements CartService{

    @Autowired
    private CartRepository cartRepository;

    public Cart create(Cart cart) {
        return cartRepository.create(cart);
    }

    public Cart read(String cartId) {
        return cartRepository.read(cartId);
    }

}
```

CartController.java

```
package com.carshop.controller;

import javax.servlet.http.HttpServletRequest;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;

@Controller
@RequestMapping("/cart")
public class CartController {
```

```
  @Autowired
  private CartService cartService;

  @GetMapping
  public String requestCartId(HttpServletRequest request) {
    String sessionid = request.getSession(true).getId();
    return "redirect:/cart/" + sessionid;

    // 경로 /cart로 요청이 들어오면 세션id 값을 가져와서 cart/세션값으로 다시호출
  }

  @PostMapping
  public @ResponseBody Cart create(@RequestBody Cart cart) {
    return cartService.create(cart);
  }
    // create() 매서드는 장바구니를 새로 생성하고 응답을 body 로 전달한다.

  @GetMapping("/{cartId}")
  public String requestCartList(@PathVariable(value="cartId") String cartId, Model model) {
    Cart cart = cartService.read(cartId);
    model.addAttribute("cart", cart);
    return "cart";
  }
    // 요청 uri 가 /cart/장바구니아이디 로 get 으로 요청되면 처리 되는 매서드 해당 장바구니 아이디 cartId 의
    // 모든 정보를 읽어서 cart 속성에 등록하고 뷰이름 cart 를 반환하므로 cart.jsp 를 호출하게 된다.

  @PutMapping("/{cartId}")
  public @ResponseBody Cart read(@PathVariable(value="cartId") String cartId) {
    return cartService.read(cartId);
  }
    // 해당 cart/cartId 값으로 요청이 되면 read() 해당 장바구니 에 등록된 모든 정보 읽어오기
}
```

car.jsp 단추 세개 추가 (제품 설명 아래쪽)

```
<p>
    <a href="#" 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>
```

cart.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>Car Detail</title>
<script>
  function removeFromCart(action) {
```

```
      document.removeForm.action = action;
      document.removeForm.submit();
      window.location.reload();
  }

  function clearCart() {
      document.clearForm.submit();
      window.location.reload();
  }
  </script>

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



  <div class="my-5">
    <div class="alert alert-dark">
      <div class="container">
        <h1>장바구니</h1>
      </div>
    </div>
  </div>

  <div class="container">
    <div class="d-grid gap-2 d-md-flex justify-content-md-end">
      <form:form name="clearForm" method="delete">
        <a href="javascript:clearCart()" class="btn btn-danger pull-left">삭제하기</a>
        <a href="#" class="btn btn-success float-right"> 주문하기</a>
      </form:form>
    </div>
    <div style="padding-top: 50px">
      <table class="table table-hover">
        <tr>
          <th>제품</th>
          <th>가격</th>
          <th>수량</th>
          <th>소계</th>
          <th>비고</th>
        </tr>
        <form:form name="removeForm" method="put">
          <c:forEach items="${cart.cartItems}" var="item">
            <tr>
              <td>${item.value.car.cid}-${item.value.car.cname}</td>
              <td>${item.value.car.cprice}</td>
              <td>${item.value.quantity}</td>
              <td>${item.value.totalPrice}</td>
              <td><a
                href="javascript:removeFromCart('../cart/remove/${item.value.car.cid}')"
                class="btn btn-danger btn-sm">삭제</a></td>
            </tr>
          </c:forEach>
        </form:form>
        <tr>
          <th></th>
          <th></th>
          <th>총액</th>
          <th>${cart.grandTotal}</th>
          <th></th>
        </tr>
      </table>

      <a href="<c:url value="/cars" />" class="btn btn-secondary">
        &laquo; 쇼핑 계속하기</a>
    </div>
```

```
    <hr>
    <%@ include file="footer.jsp"%>

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

장바구니에 제품 넣고 삭제

web.xml 에 HiddenHttpMethodFilter 를 설정

브라우저는 기본적 으로 get / post 만 지원한다. get/post 이외의 http 매서드를 사용하기 위해서 설정 ….. PUT / DELETE / PATCH

```
  <filter>
    <filter-name>httpMethodFilter</filter-name>
    <filter-class>org.springframework.web.filter.HiddenHttpMethodFilter</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>httpMethodFilter</filter-name>
    <servlet-name>appServlet</servlet-name>
  </filter-mapping>
```

Cart 클래스에 addCartItem() 매서드 추가, 선택한 제품을 장바구니로 등록하는 매서드

```
public void addCartItem(CartItem item) {
    String cid = item.getCar().getCid(); //등록할 제품 id 가져오기

    //이미 해당 제품이 등록이 되어 있는지 여부 확인
    if (cartItems.containsKey(cid)) {
      CartItem cartItem = cartItems.get(cid);//정보 가져와서
      //개수만 하나 늘려준다.
      cartItem.setQuantity(cartItem.getQuantity() + item.getQuantity());
      cartItems.put(cid, cartItem); //변경정보 저장
    } else {
      cartItems.put(cid, item); //제품 정보 저장
    }

    updateGrandTotal();  //총액 다시 계산
  }
```

CartRepository.java

```java
package com.carshop.controller;

public interface CartRepository {

    Cart create(Cart cart);
    Cart read(String cartId);
    void update(String cartId, Cart cart);
}
```

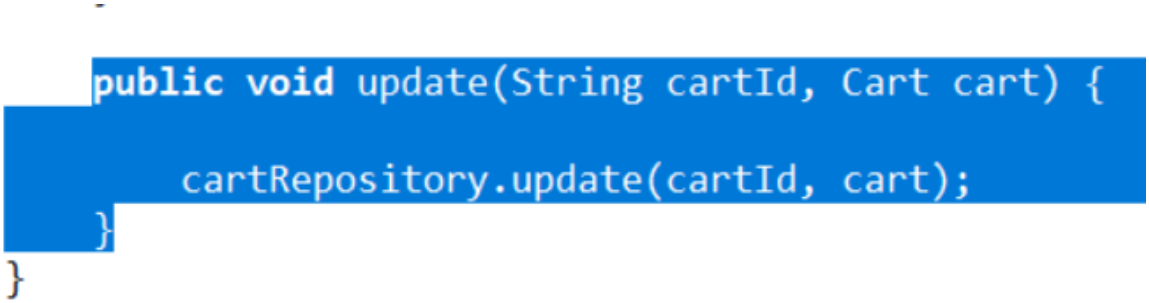
CartRepositoryImpl.java

```java
public void update(String cartId, Cart cart) {

    listOfCarts.put(cartId, cart);
}
```

CartService.java

```java
package com.carshop.controller;

public interface CartService {

    Cart create(Cart cart);
    Cart read(String cartId);
    void update(String cartId, Cart cart);
}
```

CartServiceImpl.java



```
    public void update(String cartId, Cart cart) {

        cartRepository.update(cartId, cart);
    }
}
```

CartController.java 에 addCartByNewItem 즉 장바구니에 제품 추가

```
@PutMapping("/add/{cid}")
  public void addCartByNewItem(@PathVariable String cid, HttpServletRequest request) {
    //장바구니 ID 즉 세션 ID 가져오기
    String sessionId = request.getSession(true).getId();
    Cart cart = cartService.read(sessionId);  //장바구니에 등록되어있는 모든 정보 가져오기
    if (cart == null)
      cart = cartService.create(new Cart(sessionId));

    //cid 에 대한 정보 가져오기
    CarDTO car = carService.getCarById(cid);
    //cart 아이템으로 해당 제품을 등록
    cart.addCartItem(new CartItem(car));
    //해당 장바구니에 대한 정보 갱신
    cartService.update(sessionId, cart);
  }
```

car.jsp

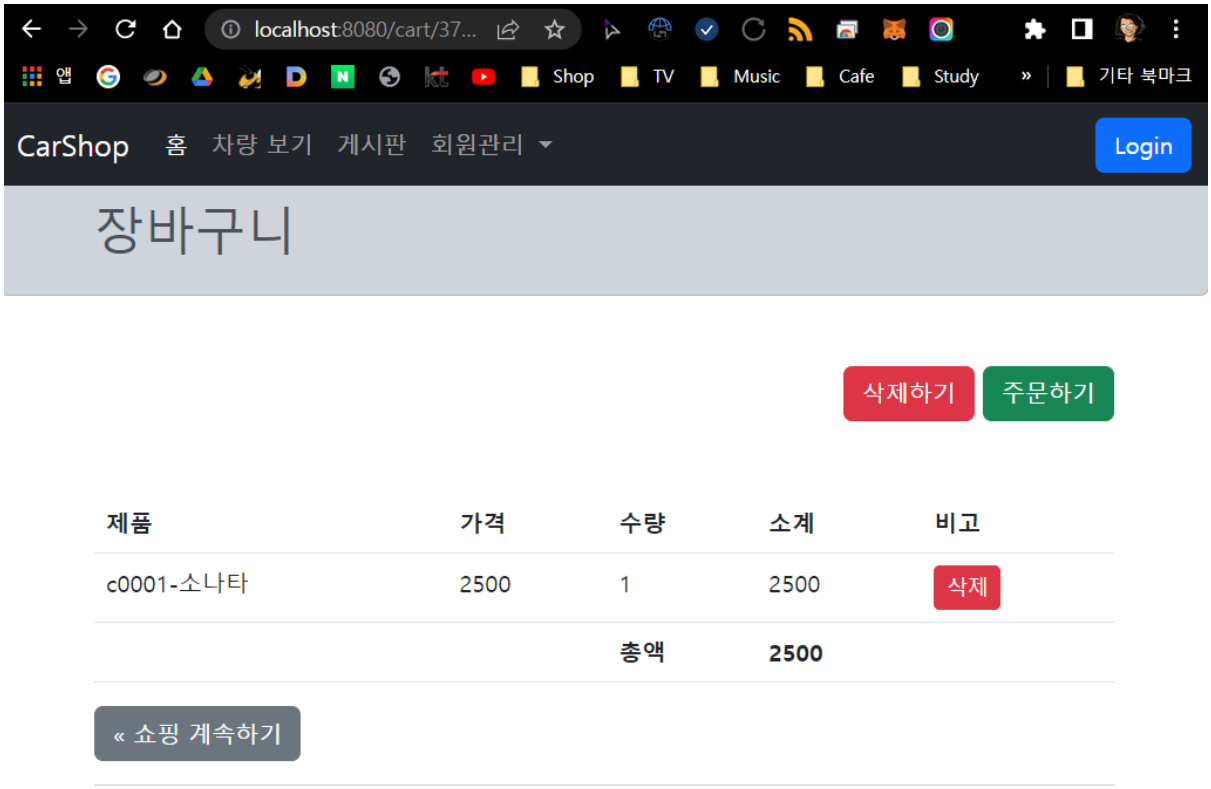
```
구매하려는 제품을 장바구니로 보낼때 스크립트를 사용하여 화면전환이 없이도
submit()을 작동 시킨다.

<script>
function addToCart(action) {
  document.addForm.action = action;
  document.addForm.submit();
  alert("제품이 장바구니에 추가되었습니다.")
}
</script>
```

```
    <p>
    <form:form name="addForm" method="put">
    <a href="javascript:addToCart('/cart/add/${car.cid}')" class="btn btn-primary">제품주문
    <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>
```

```
<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>
```

주의 !!! 폼테그 확인 !!!

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



삭제 처리

Cart.jsp 에 삭제 매서드 등록

```
public void removeCartItem(CartItem item) {
    String cid = item.getCar().getCid(); //삭제할 제품 id 가져오기
    cartItems.remove(cid); //해당 제품 삭제
    updateGrandTotal(); //총액 재계산
  }
```

CartController.java

```
@PutMapping("/remove/{cid}")
  public void removeCartByItem(@PathVariable String cid, HttpServletRequest request) {
    String sessionId = request.getSession(true).getId();
    Cart cart = cartService.read(sessionId); //장바구니에 등록되어있는 모든 정보 가져오기

    if (cart == null)
```

```
      cart = cartService.create(new Cart(sessionId));

    //cid 에 대한 정보 가져오기
    CarDTO car = carService.getCarById(cid);
    cart.removeCartItem(new CartItem(car));
    cartService.update(sessionId, cart);  //장바구니 갱신
  }
```

Cart.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>Car Detail</title>
<script>
  function removeFromCart(action) {
      document.removeForm.action = action;
      document.removeForm.submit();
      window.location.reload();
  }

  function clearCart() {
      document.clearForm.submit();
      window.location.reload();
  }
  </script>

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



  <div class="my-5">
    <div class="alert alert-dark">
      <div class="container">
        <h1>장바구니</h1>
      </div>
    </div>
  </div>

  <div class="container">
    <div class="d-grid gap-2 d-md-flex justify-content-md-end">
      <form:form name="clearForm" method="delete">
        <a href="javascript:clearCart()" class="btn btn-danger pull-left">삭제하기</a>
        <a href="#" class="btn btn-success float-right"> 주문하기</a>
      </form:form>
    </div>
    <div style="padding-top: 50px">
      <table class="table table-hover">
        <tr>
          <th>제품</th>
          <th>가격</th>
          <th>수량</th>
          <th>소계</th>
```

```
          <th>비고</th>
        </tr>
        <form:form name="removeForm" method="put">
          <c:forEach items="${cart.cartItems}" var="item">
            <tr>
              <td>${item.value.car.cid}-${item.value.car.cname}</td>
              <td>${item.value.car.cprice}</td>
              <td>${item.value.quantity}</td>
              <td>${item.value.totalPrice}</td>
              <td><a
                href="javascript:removeFromCart('../cart/remove/${item.value.car.cid}')"
                class="btn btn-danger btn-sm">삭제</a></td>
            </tr>
          </c:forEach>
        </form:form>
        <tr>
          <th></th>
          <th></th>
          <th>총액</th>
          <th>${cart.grandTotal}</th>
          <th></th>
        </tr>
      </table>

      <a href="<c:url value="/cars" />" class="btn btn-secondary">
        &laquo; 쇼핑 계속하기</a>
    </div>
    <hr>
    <%@ include file="footer.jsp"%>

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

!! 주의 - 삭제 기능은 내장 브라우저에서 원활하게 작동, 외부 크롬에서는 작동하지 않으나 서버에 올리면 잘 작동하므로 별 문제가 없다.

장바구니 전체 한번에 제거

repository

```java
package com.carshop.controller;

public interface CartRepository {

    Cart create(Cart cart);
    Cart read(String cartId);
    void update(String cartId, Cart cart);
    void delete(String cartId);
}
```

```java
    public void delete(String cartId) {

        listOfCarts.remove(cartId);
    }
```

service

```java
package com.carshop.controller;

public interface CartService {

    Cart create(Cart cart);
    Cart read(String cartId);
    void update(String cartId, Cart cart);
    void delete(String cartId);
}
```

```java
    public void delete(String cartId) {
        cartRepository.delete(cartId);
    }
```

controller

```
@DeleteMapping("/{cartId}")
public void deleteCartList(@PathVariable(value="cartId") String cartId) {
    cartService.delete(cartId);
}
```

view

```
<div class="container">
    <div class="d-grid gap-2 d-md-flex justify-content-md-end">
        <form:form name="clearForm" method="delete">
            <a href="javascript:clearCart()" class="btn btn-danger pull-left">삭제하
            <a href="#" class="btn btn-success float-right"> 주문하기</a>
        </form:form>
    </div>
```

```
function clearCart() {
    document.clearForm.submit();
    window.location.reload();
}
</script>
```

Bug Report

1. 장바구니에 제품 추가하고 404로 넘어가는 문제

-보류-

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

        </div>
    </div>




</body>

<iframe name="cart" style="display: none;"></iframe>

<script type="text/javascript">

//구매하려는 제품을 장바구니로 보낼 때 JS를 이용하여 화면전환 없이 submit()을 실행한다.
function addToCart(action) {
    document.addForm.action = action;
    document.addForm.submit();
    alert("제품이 장바구니에 추가되었습니다.");
}
```

1. 장바구니에 같은 제품 추가시 개발 가격 재계산 문제 (해결)

```
public void setQuantity(int quantity) {
    this.quantity = quantity;
    this.updateTotalPrice();
  }
```

CartItem.java

```
19    public void setQuantity(int quantity) {
20        this.quantity = quantity;
21        this.updateTotalPrice();
22    }
```

가격	수량	소계	비고
2500	2	5000	삭제
2000	1	2000	삭제
	총액	**7000**	

삭제는 이클립스 내부 브라우저에서 테스트 할것

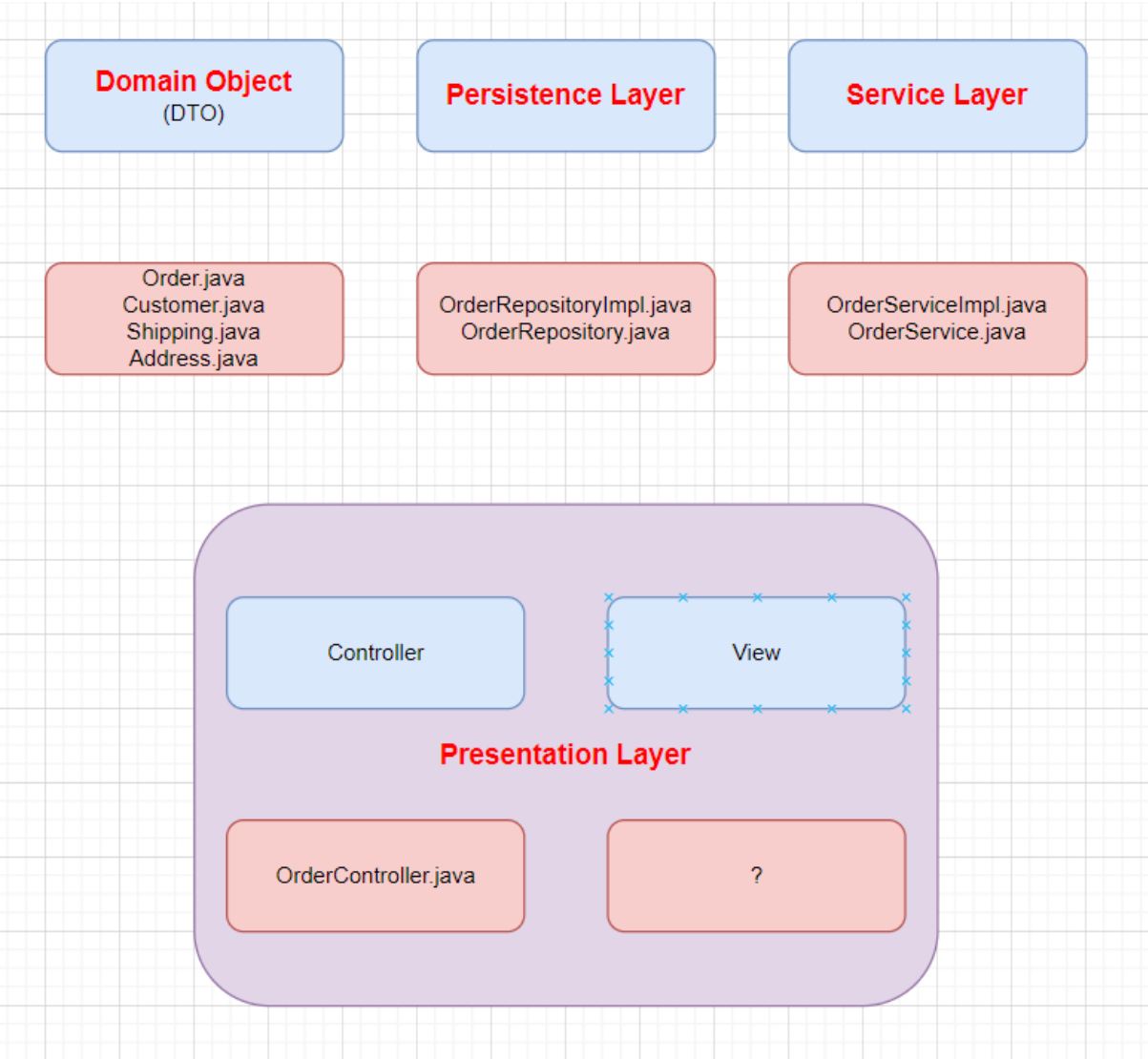
외부 크롬에서는 동작하지 않음

Spring Web Flow

웹 페이지 구성이 복잡한 사이트를 개발할 때 페이지들의 흐름을 추적하고 관리할수 있는 기능

주문 처리 흐름 : 주문신청 → 주소입력 → 결제 → 주문완료

- 웹 페이지 흐름을 깔끔하게 한눈에 파악할수 있다.
- 스프링 외의 다른 프레임워크와도 연동해서 사용할수 있다. 즉 스프링에 포함된것 은 아니다.
- 특정 컨트롤러를 사용하지 않고 적절한 흐름을 관리 할수 있다.
- 사이트가 복잡할때 흐름을 관리하여 결과적으로 사용하기 쉽게 만들어 준다.



CarDTO, Cart, CartItem 클래스를 모두 수정

- 위 모든 클래스를 Serializable 인터페이스의 구현체로 수정한다.

```
import java.io.Serializable;

public class CarDTO implements Serializable {
```

```java
import java.io.Serializable;

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