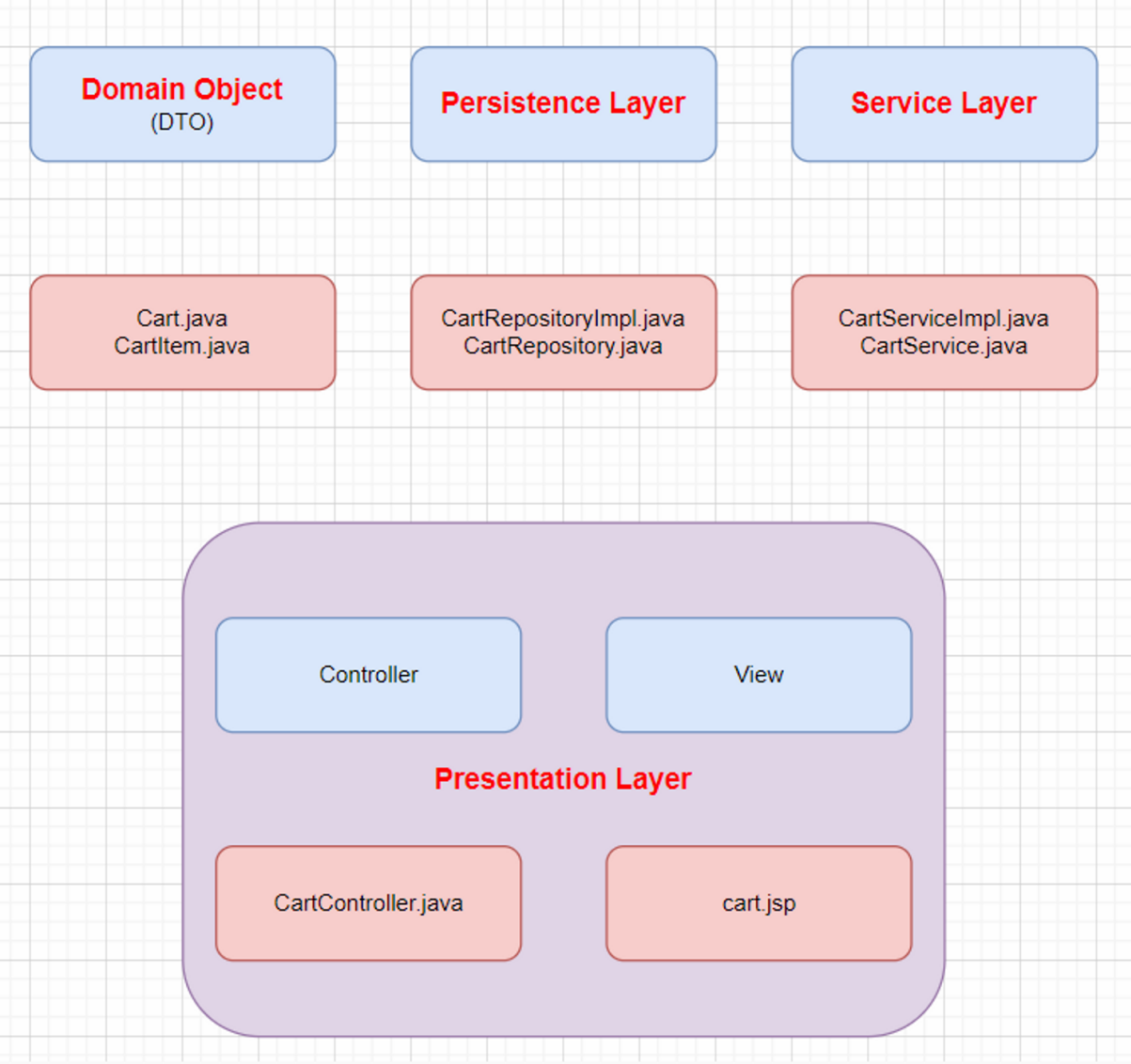
230308 스프링



CartController

```
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 세션 id값을 가져와서 cart/세션값으로 다시 호출
  }

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

  // 요청 uri가 /cart/장바구니아이디로 get으로 요청되면 처리되는 메서드는 해당 장바구니의
  // 아이디 cartId의 모든 정보를 읽어서 cart 속성에 등록하고 뷰 cart.jsp를 반환
  @GetMapping("/{cartId}")
  public String requestCartList(@PathVariable(value="cartId") String cartId, Model mod
el) {
    Cart cart = cartService.read(cartId);
    model.addAttribute("cart", cart);
    return "cart";
  }

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

장바구니에 제품 넣고 삭제

web.xml에 HiddenHttpMethodFilter를 설정

브라우저는 기본적으로 get, post만 지원한다.

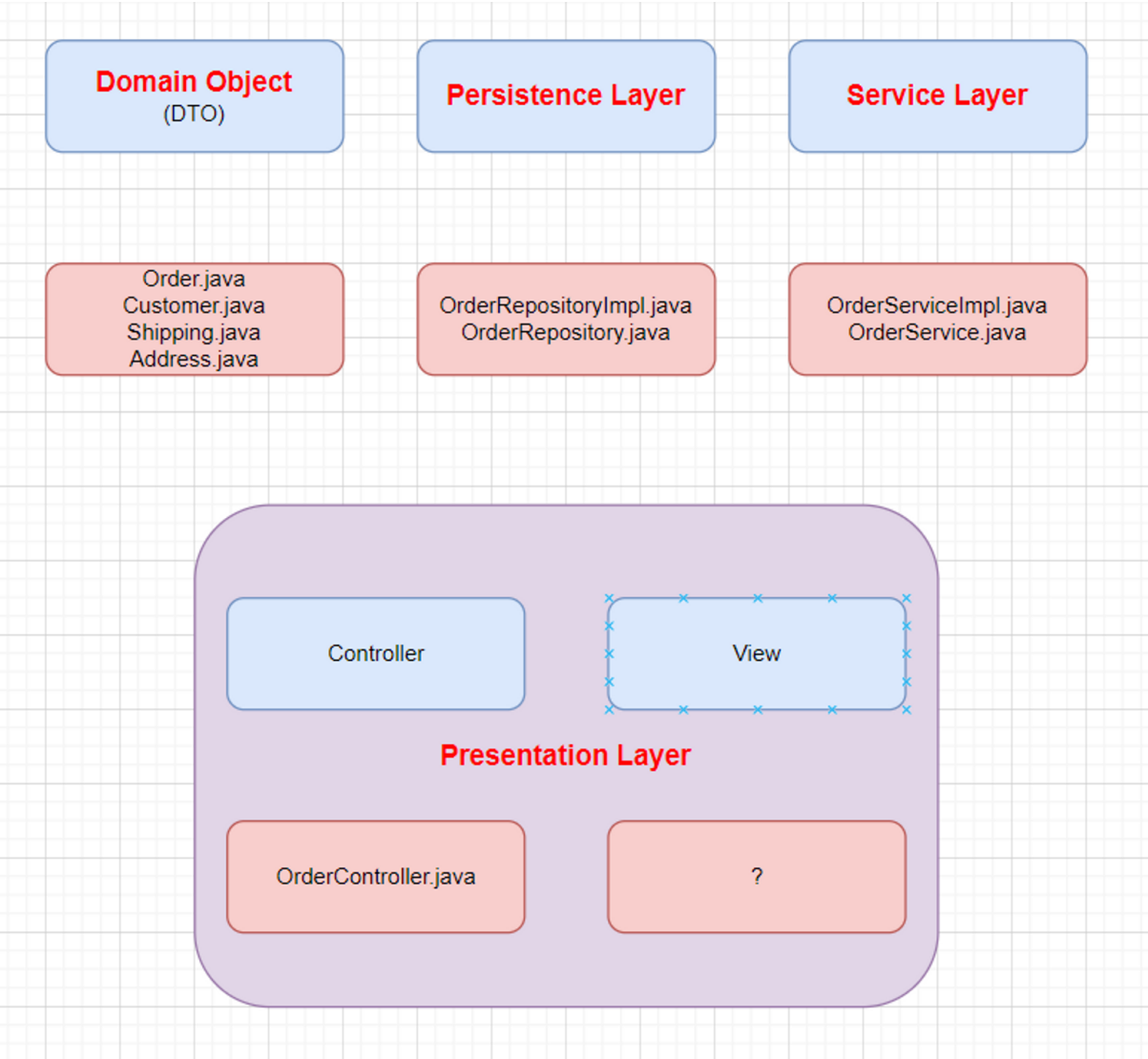
get, post 이외의 http 메서드를 사용하기 위해서 설정

Spring Web Flow

- 웹 페이지의 구성이 복잡한 사이트를 개발할 때 페이지들의 흐름을 추적하고 관리할 수 있는 기능

주문 신청 → 주소 입력 →결제 → 주문완료

- 웹 페이지의 흐름을 깔끔하게 파악 가능
- 스프링 외의 다른 프레임워크와도 연동해서 사용 가능.
- 특정 컨트롤러를 사용하지 않고 적절한 흐름 관리 가능
- 사이트가 복잡할 때 흐름을 관리하여 결과적으로 사용하기 쉽게 만들어줌.



CarDTO, Cart, CartItem 클래스를 모두 수정

- 위 클래스를 Serializalbe 인터페이스의 구현체로 모두 수정