Spring Day 6 230307

war를 zip으로 변환한 파일이 아닌 전체를 그대로 zip으로 압축한 파일

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/b356993e-d793-468d-8314-159b23632207/CarShop6.zip

따라서 import 로 읽지 말고 open project from file system 으로 읽을 것

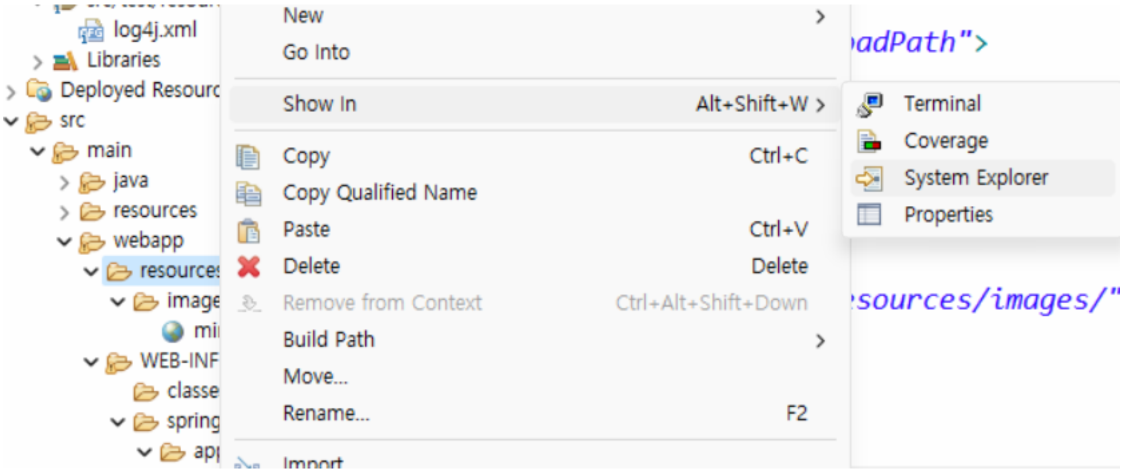
1. 실제 서버에서의 위치 찾기 cars.jsp 에서 임시 확인

   <%=request.getSession().getServletContext().getRealPath("/") %>

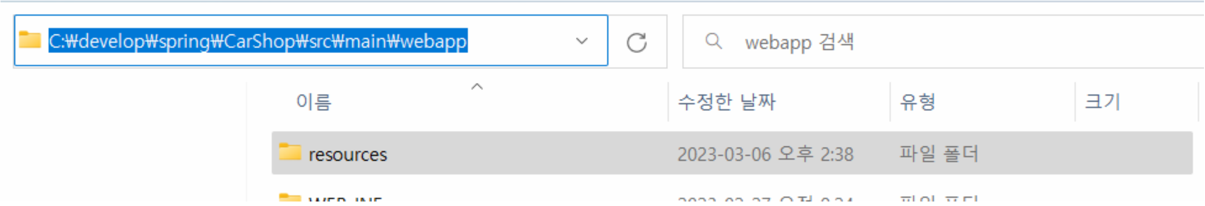
   C:\develop\spring\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\CarShop\

2. servlet-context.xml에 파일 저장에 필요한 코드를 추가

```
<!-- 업로드 패스 설정 -->
  <beans:bean class="java.lang.String" id="uploadPath">
    <beans:constructor-arg
        value="C:\develop\spring\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\CarShop\resource" />
      <!--value="C:\Develop\spring\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\BookMarket\resources" />-->
  </beans:bean>

  <!-- 일반 파일 업로드 경로 -->
  <resources mapping="/images/**" location="/resources/images/" />
```



webapp - resources - 우클릭 - show In - System Explorer



C:₩develop₩spring₩CarShop₩src₩main₩webapp

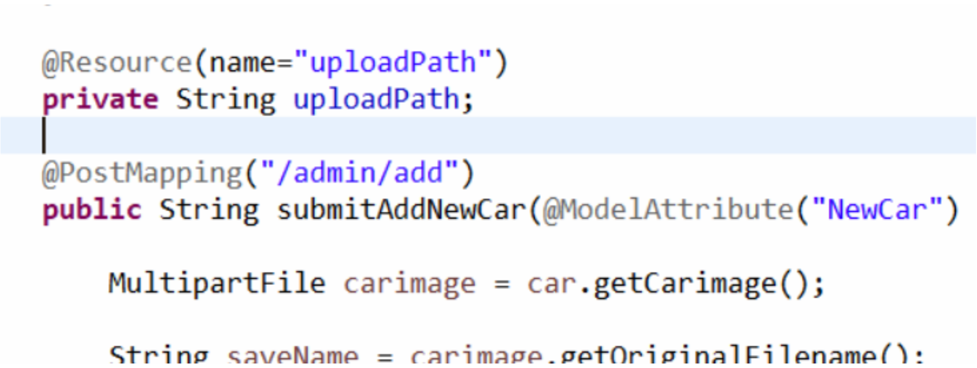
이름	수정한 날짜	유형	크기
resources	2023-03-06 오후 2:38	파일 폴더	
WEB-INF	2023-02-27 오전 9:24	파일 폴더	

로컬에서의 작업 위치 - C:\develop\spring\CarShop\src\main\webapp

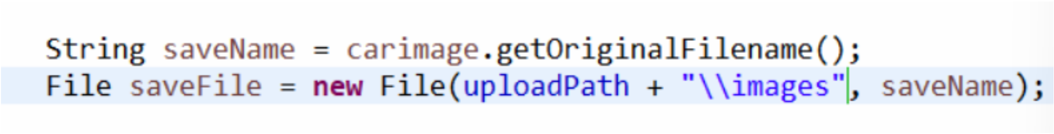
실제 운영 서버에서의 위치 (카페24 등의 서버에서는 주소형태가 운영체제에 따라 달라진다)-
C:\develop\spring\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\CarShop\

```
<!-- 업로드 패스 설정 -->
  <beans:bean class="java.lang.String" id="uploadPath">
    <beans:constructor-arg
      value="C:\develop\spring\CarShop\src\main\webapp\resources" />
  </beans:bean>
        <!--value="C:\develop\spring\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\CarShop\resource" />-->
  <!-- 일반 파일 업로드 경로 -->
  <resources mapping="/images/**" location="/resources/images/" />용
```

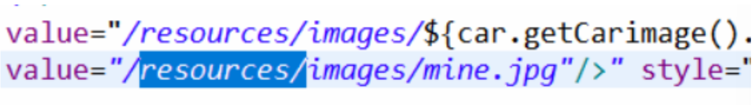
3. 사용



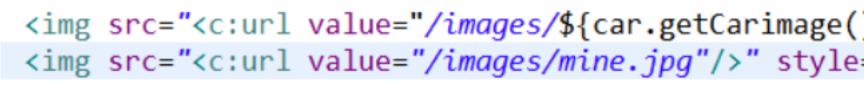
```
@Resource(name="uploadPath")
private String uploadPath;

@PostMapping("/admin/add")
public String submitAddNewCar(@ModelAttribute("NewCar")

    MultipartFile carimage = car.getCarimage();

    String saveName = carimage.getOriginalFilename();
```



```
String saveName = carimage.getOriginalFilename();
File saveFile = new File(uploadPath + "\\images", saveName);
```



```
value="/resources/images/${car.getCarimage().
value="/resources/images/mine.jpg"/>" style="
```



```
<img src="<c:url value="/images/${car.getCarimage(
<img src="<c:url value="/images/mine.jpg"/>" style
```

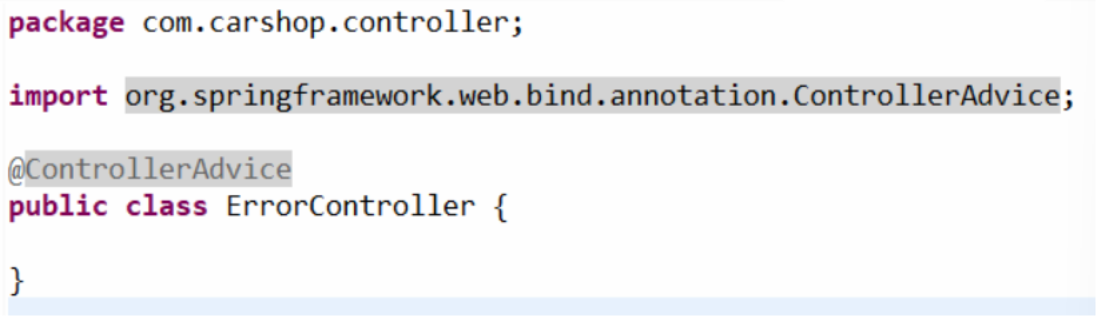
예외 처리

!! 중요.....내장 브라우저에서는 예외 처리가 확인이 잘 된다. 크롬에서 테스트 하자. !!

2가지…(4가지)

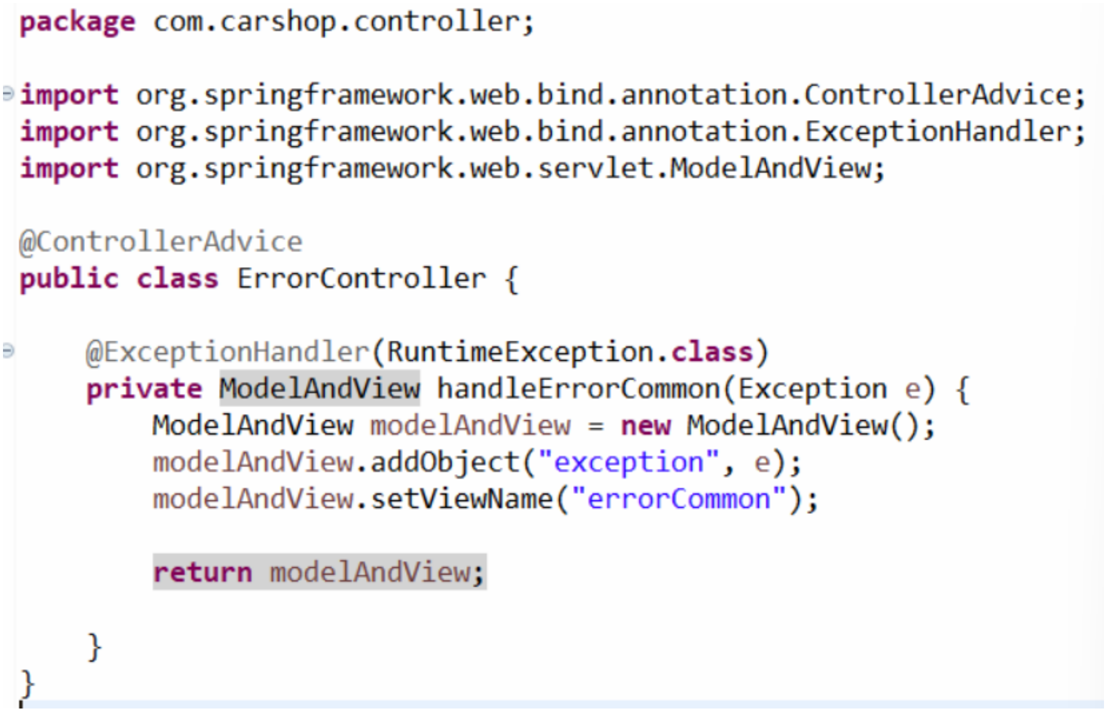
1. (404 처리는 안된다.) 대부분의 예외처리 방법..

   ErrorController.java



```
package com.carshop.controller;

import org.springframework.web.bind.annotation.ControllerAdvice;

@ControllerAdvice
public class ErrorController {

}
```

```
package com.carshop.controller;

import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.bind.annotation.ExceptionHandler;
import org.springframework.web.servlet.ModelAndView;

@ControllerAdvice
public class ErrorController {

  @ExceptionHandler(RuntimeException.class)
  private ModelAndView handleErrorCommon(Exception e) {
    ModelAndView modelAndView = new ModelAndView();
    modelAndView.addObject("exception", e);
    modelAndView.setViewName("errorCommon");

    return modelAndView;

  }
}
```



```java
package com.carshop.controller;

import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.bind.annotation.ExceptionHandler;
import org.springframework.web.servlet.ModelAndView;

@ControllerAdvice
public class ErrorController {

    @ExceptionHandler(RuntimeException.class)
    private ModelAndView handleErrorCommon(Exception e) {
        ModelAndView modelAndView = new ModelAndView();
        modelAndView.addObject("exception", e);
        modelAndView.setViewName("errorCommon");

        return modelAndView;

    }
}
```

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
${exception}
<br>
errorCommon.jsp
</body>
</html>
```

2. 404 처리

web.xml

```
<error-page>

      <error-code>404</error-code>

      <location>/WEB-INF/views/error.jsp</location>

  </error-page>
```

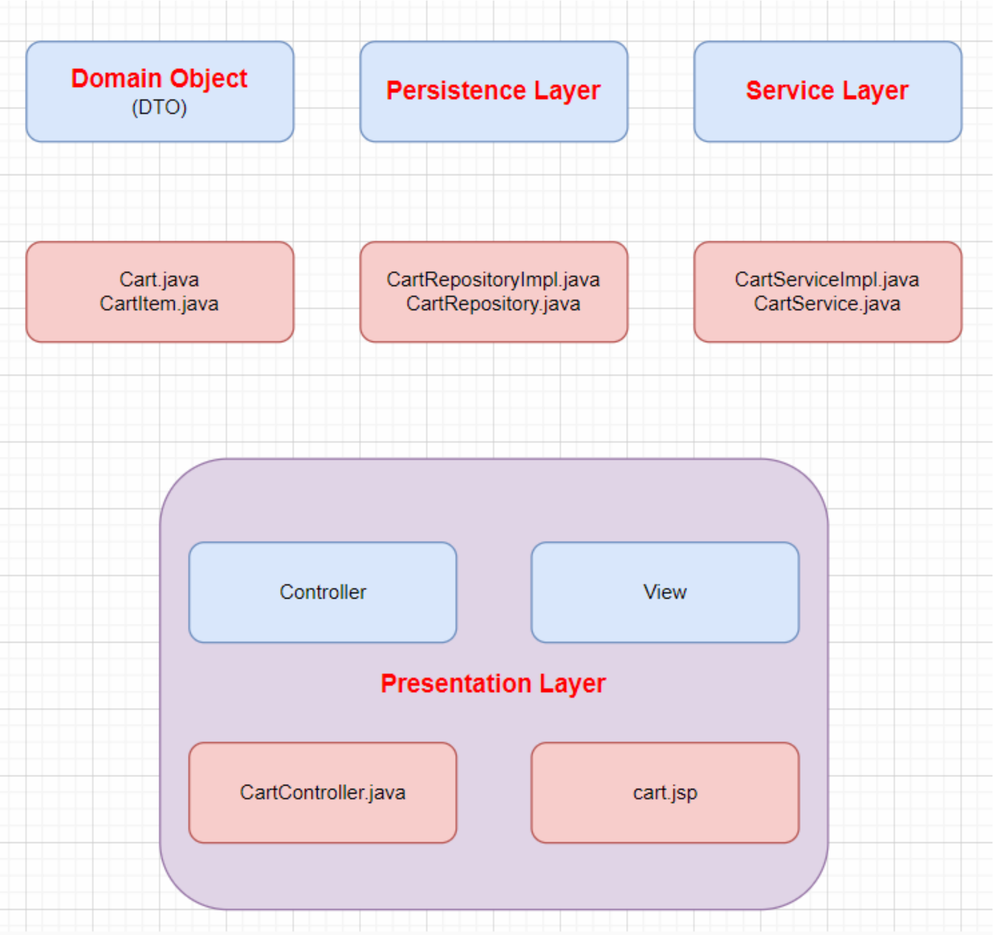
에러

[Tomcat 에러]Several ports (8005, 8080, 8009) required by Tomcat v9.0 Server at localhost are already in use. The server may already be running ir

톰캣을 구동시켰을 때 생긴 포트 오류이다.나는 이클립스가 비정상적으로 종료가 된 후에 톰캣을 구동시켰을 때 발생했다. Several ports (8005, 8080, 8009) required by Tomcat v process may be using the port. To start this server you will need to stop the other process or change the port number(s). Tomcat이 사용하고 있는 기본 포트(8080, 8009, 8

https://to-dy.tistory.com/59

장바구니 처리



CartItem.java

```
package com.carshop.controller;

public class CartItem {

  private CarDTO car;    //자동차
  private int quantity;  //자동차 개수
  private int totalPrice;//자동차 가격


  public CarDTO getCar() {
    return car;
  }
  public void setCar(CarDTO car) {
    this.car = car;
  }
  public int getQuantity() {
    return quantity;
  }
  public void setQuantity(int quantity) {
    this.quantity = quantity;
  }
  public int getTotalPrice() {
    return totalPrice;
  }
  public void setTotalPrice(int totalPrice) {
    this.totalPrice = totalPrice;
  }
  public CartItem() {
    super();
    // TODO Auto-generated constructor stub
  }
  public CartItem(CarDTO car) {
    super();
    this.car = car;
    this.quantity = 1;
    this.totalPrice = Integer.parseInt(car.getCprice());
  }


  @Override
  public int hashCode() {
    final int prime = 31;
    int result = 1;
    result = prime * result + ((car == null) ? 0 : car.hashCode());
    result = prime * result + quantity;
    result = prime * result + totalPrice;
    return result;
  }
  @Override
  public boolean equals(Object obj) {
    if (this == obj)
      return true;
    if (obj == null)
      return false;
    if (getClass() != obj.getClass())
      return false;
    CartItem other = (CartItem) obj;
    if (car == null) {
      if (other.car != null)
        return false;
    } else if (!car.equals(other.car))
      return false;
    if (quantity != other.quantity)
      return false;
    if (totalPrice != other.totalPrice)
      return false;
    return true;
  }



  public void updateTotalPrice() {

    int price = Integer.parseInt(car.getCprice());
    totalPrice = price * this.quantity;
  }




}
```

Cart.java

```
package com.carshop.controller;

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

public class Cart {

  private String cartId;
  private Map<String, CartItem> cartItems;
  private int grandTotal;



  public Cart() {
    cartItems = new HashMap<String, CartItem>();
  }
  public Cart(String cartId) {
    this();
    this.cartId = cartId;
  }



  public String getCartId() {
    return cartId;
  }



  public void setCartId(String cartId) {
    this.cartId = cartId;
  }



  public Map<String, CartItem> getCartItems() {
    return cartItems;
  }



  public void setCartItems(Map<String, CartItem> cartItems) {
    this.cartItems = cartItems;
  }



  public int getGrandTotal() {
    return grandTotal;
  }



  public void setGrandTotal(int grandTotal) {
    this.grandTotal = grandTotal;
  }



  @Override
  public int hashCode() {
    final int prime = 31;
    int result = 1;
    result = prime * result + ((cartId == null) ? 0 : cartId.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;
    Cart other = (Cart) obj;
    if (cartId == null) {
      if (other.cartId != null)
        return false;
    } else if (!cartId.equals(other.cartId))
      return false;
    return true;
  }


  public void updateGrandTotal() {
    grandTotal = 0;
    for(CartItem item : cartItems.values()) {
      grandTotal = grandTotal + item.getTotalPrice();
    }
  }




}
```

CartRepository.java

```
package com.carshop.controller;

public interface CartRepository {

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

CartRepositoryImpl.java

```
package com.carshop.controller;

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

import org.springframework.stereotype.Repository;

@Repository
public class CartRepositoryImpl implements CartRepository {

  private Map<String, Cart> listOfCarts;

  public CartRepositoryImpl() {
    listOfCarts = new HashMap<String, Cart>();
  }


  @Override
  public Cart create(Cart cart) {
    if (listOfCarts.keySet().contains(cart.getCartId())) {
      throw new IllegalArgumentException(
          String.format("장바구니를 생성할수 없습니다.장바구니 id가 이미 존재합니다.",
              cart.getCartId()));
    }

    listOfCarts.put(cart.getCartId(), cart);

    return cart;
  }

  @Override
  public Cart read(String cartId) {

    return listOfCarts.get(cartId);
  } // cartId 를 이용하여 장바구니에 등록된 모든 정보를 가져온다.

}
```

CartServiceImpl.java

```
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);
  }
}
```

CartService.java

```
package com.carshop.controller;

public interface CartService {

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