

# Spring Day 6

## File Upload

servlet-context.xml

```
<!-- 업로드 패스 설정 -->
<beans:bean class="java.lang.String" id="uploadPath">
  <beans:constructor-arg
    value="/Users/shimhyunmin/develop/spring/CarShop/src/main/webapp/resources" />
</beans:bean>

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

## 실제 서버 위치

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

| /Users/shimhyunmin/develop/spring/.metadata/.plugins/org.eclipse.wst.server.core/tmp0/wtpwebapp

## 로컬 위치

| /Users/shimhyunmin/develop/spring/CarShop/src/main/webapp/resources

## controller

macOS 기준 슬래시 주의

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

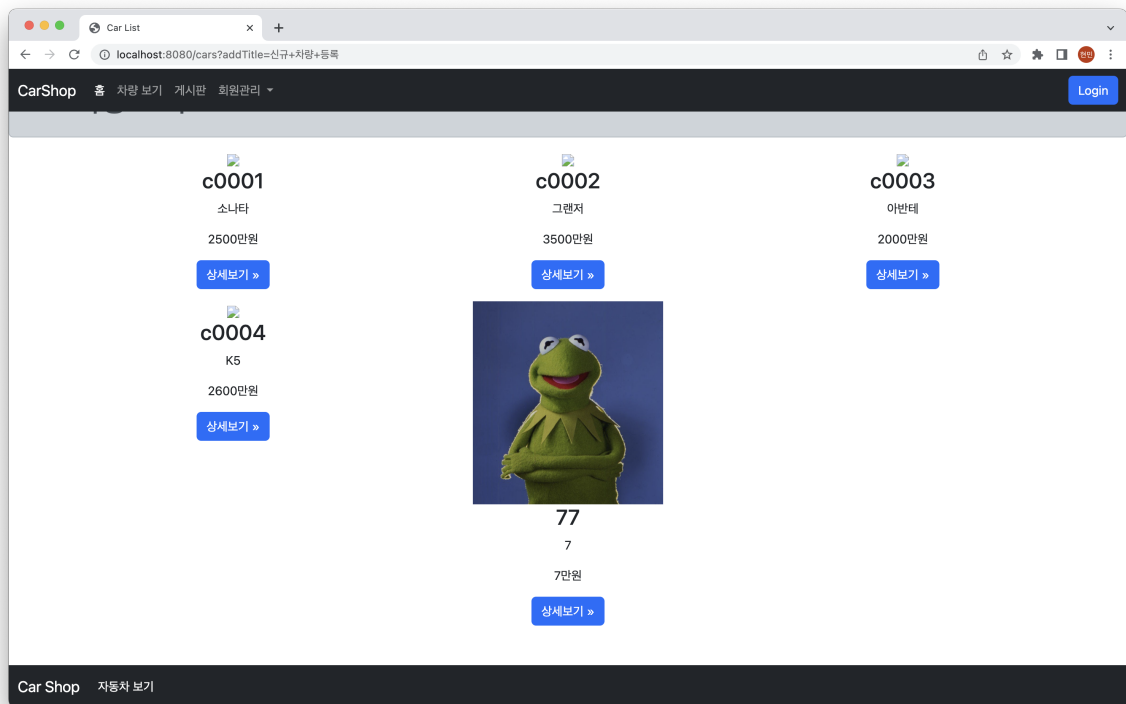
@PostMapping("/admin/add")
public String submitAddNewCar(@ModelAttribute("NewCar") CarDTO car) {

    MultipartFile carimage = car.getCimage();

    String saveName = carimage.getOriginalFilename();
    File saveFile = new File(uploadPath + "/images", saveName); // macOS 기준 슬래시

    if (carimage != null && !carimage.isEmpty()) {
        try {
            carimage.transferTo(saveFile);
        } catch (Exception e) {
            throw new RuntimeException("차량 이미지 업로드가 실패했습니다.");
        }
    }

    carService.setNewCar(car);
    return "redirect:/cars";
}
```



## Exception

### 404 제외 처리

ErrorHandler.java

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

        return modelAndView;
    }
}
```

### 404 처리

web.xml

```
<error-page>
  <error-code>404</error-code>
  <location>/WEB-INF/views/error.jsp</location>
</error-page>
```

## 장바구니

CartItem.java

```

package com.carshop.controller;

import com.carshop.car.CardTO;

import lombok.AllArgsConstructor;
import lombok.Data;

@AllArgsConstructor
@Data
public class CartItem {
    private CardTO car;    //자동차
    private int quantity;  //자동차 개수
    private int totalPrice; //자동차 가격

    public CartItem() {
        super();
    }

    public CartItem(CardTO car) {
        this.car = car;
        this.quantity = 1;
        this.totalPrice = Integer.parseInt(car.getCprice());
    }

    @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;
    }

    @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;
    }

    public void updateTotalPrice() {
        totalPrice = Integer.parseInt(this.car.getCprice())*this.quantity;
    }
}

```

## Cart.java

```

package com.carshop.controller;

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

import lombok.AllArgsConstructor;
import lombok.Data;

@AllArgsConstructor
@Data
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;
}

@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;
}

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

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 생성할 수 없습니다. 장바구니 Id가 이미 존재 합니다.", cart.getCartId()));
        }
        listOfCarts.put(cart.getCartId(), cart);
        return cart;
    }

    @Override
    public Cart read(String cartId) {

```

```
        return list0fCarts.get(cartId);  
    }  
}
```

#### CartService.java

```
package com.carshop.controller;  
  
public interface CartService {  
    Cart create(Cart cart);  
    Cart read(String cartId);  
}
```

#### 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);  
    }  
}
```