



# Spring Football Shopping Mall

with Mybatis

안지혁

# Index



01

서론

프로젝트 목적

프로젝트 개요

02

본론

프로젝트 구성도

프로젝트 기능

프로젝트 주요 기술

03

결론

개선 사항

프로젝트 후기

- Spring Mybatis를 활용한 쇼핑몰 제작
- Spring 3-tier 구조의 이해 및 활동
- Spring IoC, DI, AOP의 개념 이해 및 활용
- Spring Mybatis 연동을 통한 연속성 구현
- Tiles framework를 사용하여 화면 구현

- 메인 페이지에 등록된 상품이 베스트 셀러, 스테디셀러, 신규등록 별로 나타나도록 함
- 각 상품은 상세 보기 페이지로 연결
- 가입 회원만 상세 보기 페이지에서 상품 구매, 장바구니 담기 기능
- 장바구니에서 주문이 가능하도록 구현
- 마이페이지에서 자신의 정보 확인 및 수정 가능, 주문 정보 확인 가능
- 관리자는 상품등록, 주문 회원 관리가 가능하도록 구현
- 전체 프로젝트 구조는 Controller, Service, Repository의 3tier 구조로 구현
- Repository 영역은 Mybatis를 연동하여 구현

# Index



01

서론

프로젝트 목적

프로젝트 개요

02

본론

프로젝트 구성도

프로젝트 기능

프로젝트 주요 기술

03

결론

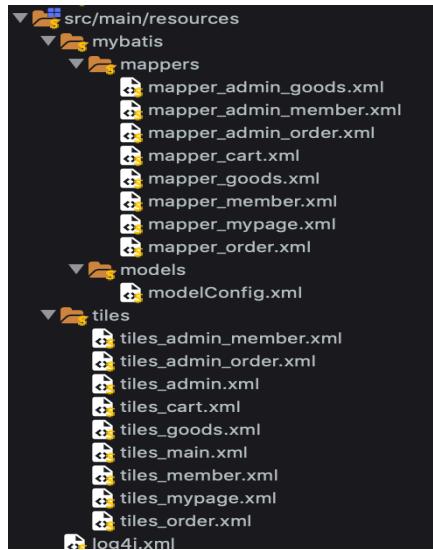
개선 사항

프로젝트 후기

```
▼ SpringFootBallMall [SpringFootBallMall master]
  ▼ src/main/java
    ▶ com.shoppingmall.admin.goods.controller
    ▶ com.shoppingmall.admin.goods.dao
    ▶ com.shoppingmall.admin.goods.service
    ▶ com.shoppingmall.admin.member.controller
    ▶ com.shoppingmall.admin.member.dao
    ▶ com.shoppingmall.admin.member.service
    ▶ com.shoppingmall.admin.order.controller
    ▶ com.shoppingmall.admin.order.dao
    ▶ com.shoppingmall.admin.order.service
    ▶ com.shoppingmall.cart.controller
    ▶ com.shoppingmall.cart.dao
    ▶ com.shoppingmall.cart.service
    ▶ com.shoppingmall.cart.vo
    ▶ com.shoppingmall.common.base
    ▶ com.shoppingmall.common.file
    ▶ com.shoppingmall.common.interceptor
    ▶ com.shoppingmall.goods.controller
    ▶ com.shoppingmall.goods.dao
    ▶ com.shoppingmall.goods.service
    ▶ com.shoppingmall.goods.vo
    ▶ com.shoppingmall.main
    ▶ com.shoppingmall.member.controller
    ▶ com.shoppingmall.member.dao
    ▶ com.shoppingmall.member.service
    ▶ com.shoppingmall.member.vo
    ▶ com.shoppingmall.mypage.controller
    ▶ com.shoppingmall.mypage.dao
    ▶ com.shoppingmall.mypage.service
    ▶ com.shoppingmall.mypage.vo
    ▶ com.shoppingmall.order.controller
    ▶ com.shoppingmall.order.dao
    ▶ com.shoppingmall.order.service
    ▶ com.shoppingmall.order.vo
```

- controller : client에게 요청을 받아 요청에 맞는 비지니스(service) 영역으로 전달
- service : 비즈니스 로직 구현 및 트랜잭션처리
- dao : 데이터 접근을 목적하는 객체 DB를 사용해 데이터를 조회하거나 조작하는 기능을 전담
- vo : DB 테이블의 컬럼과 ORM을 위한 클래스

# 프로젝트 구성도

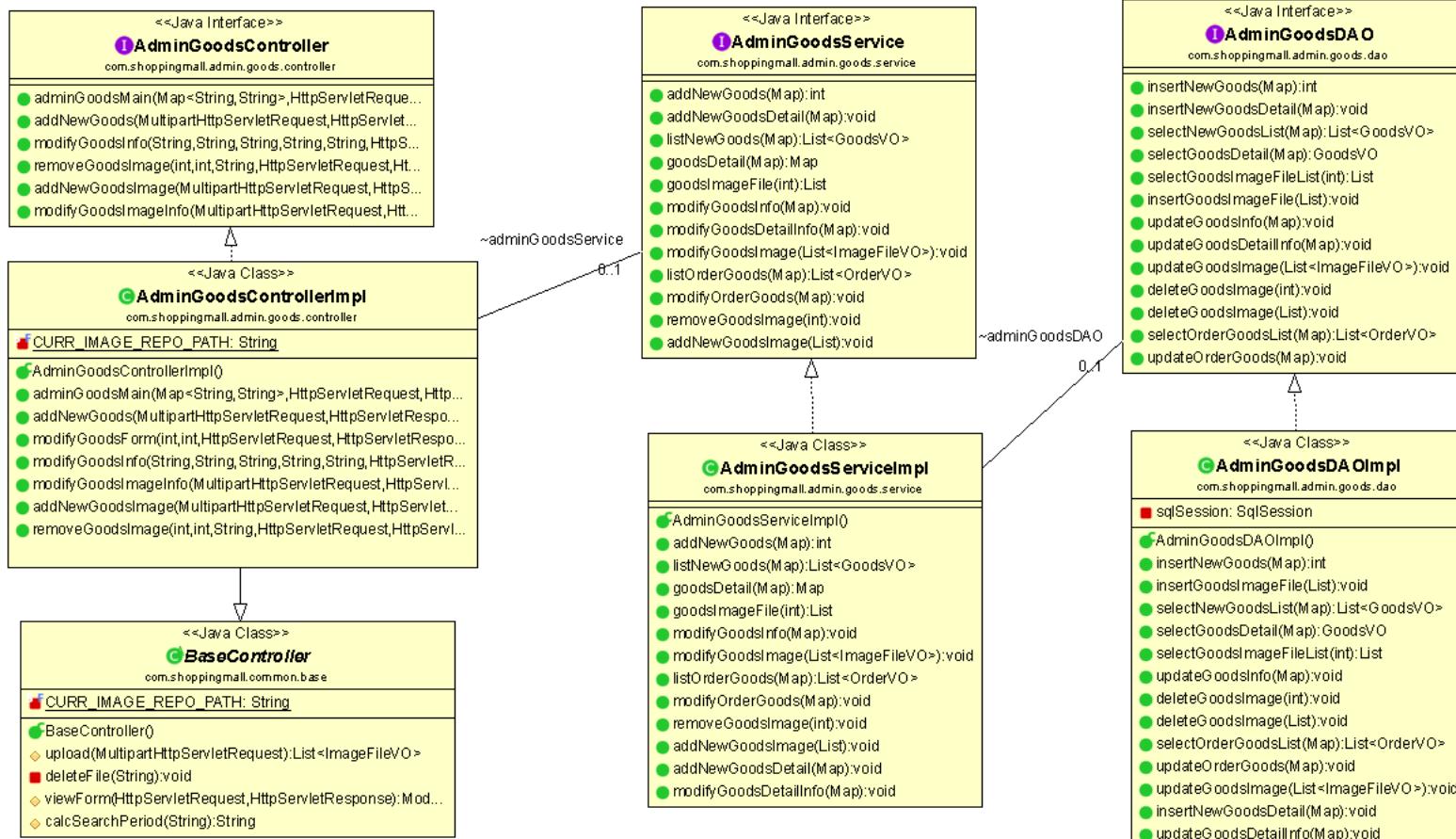


- **mappers** : Mybatis 사용 시 각 기능별로 나눈 mapper파일들(DAO)에서 호출
- **models** : mapper 파일에서 참조하는 model 클래스들의 alias를 설정
- **titles** : view단(jsp)의 탑, 사이드 메뉴, 하단, 메인 등을 페이지 include 방식으로 나누는 기존 구조를 쉽게 적용하기 위한 템플릿 프레임워크

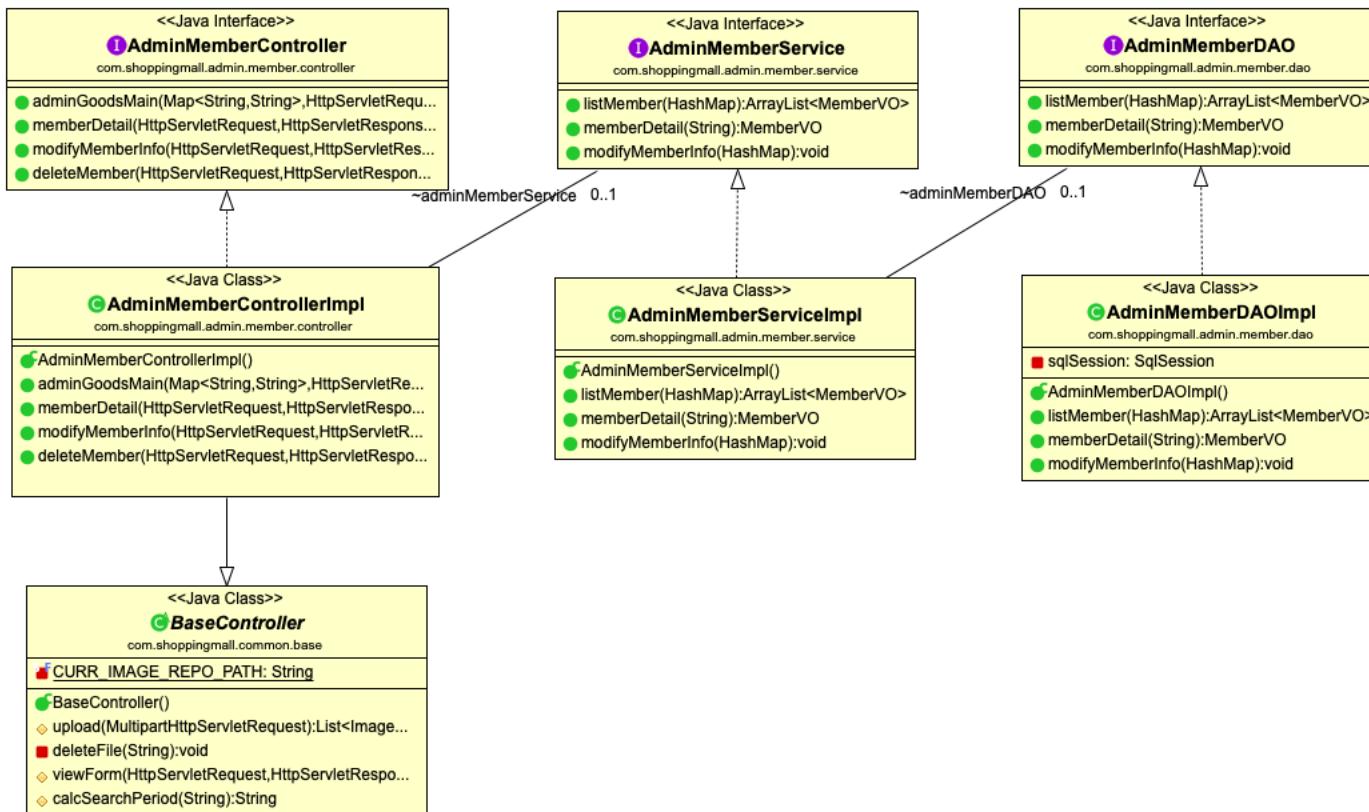


## view 단에서 사용하는 jsp 파일들

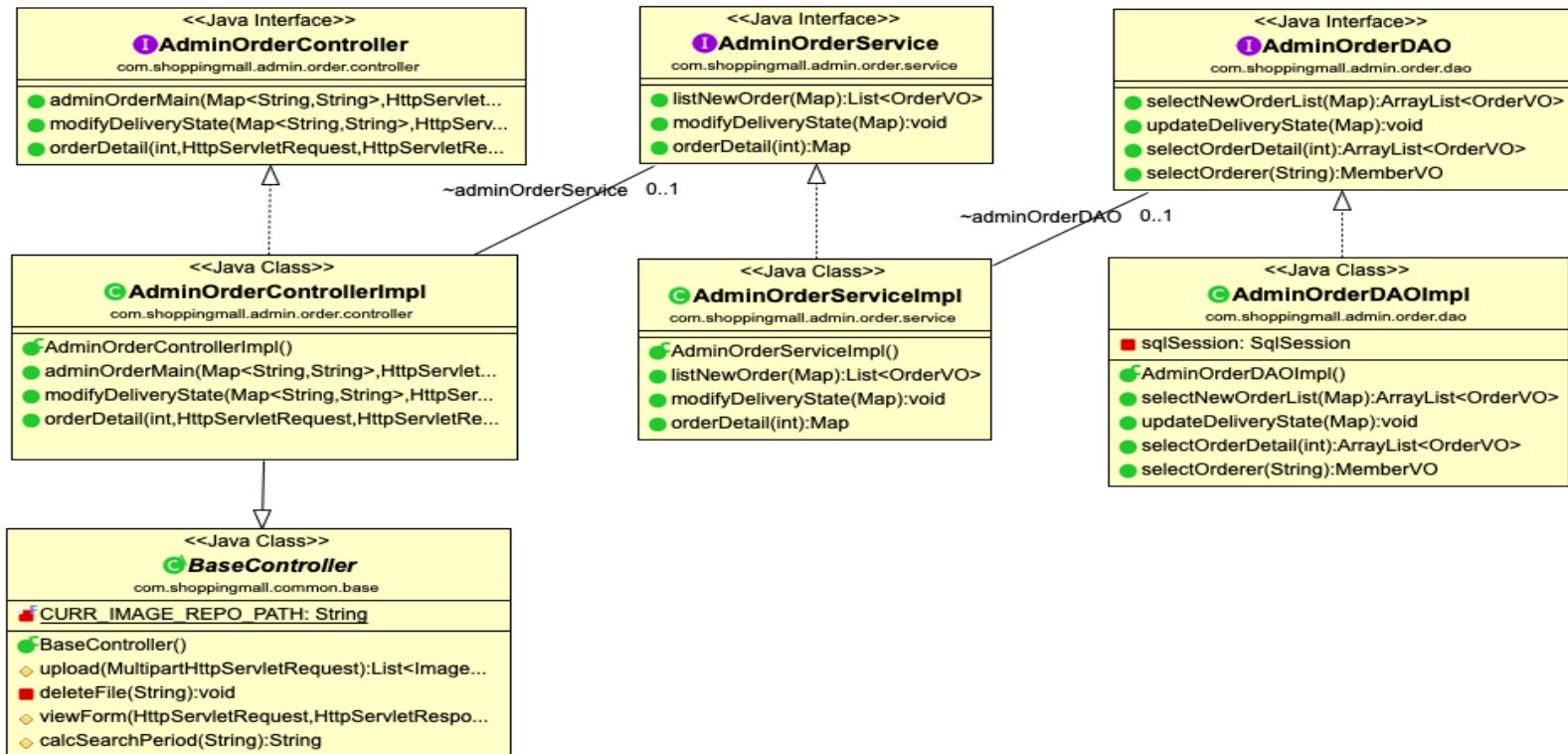
- admin(goods, member, order) : 관리자 사용 페이지
- cart : 장바구니 페이지
- goods : 상품 상세, 검색 페이지
- main : 메인화면 페이지
- member : 회원가입, 로그인 페이지
- mypage : 회원 개인 정보관리, 주문 관리 페이지
- order : 주문 관련 페이지



adminGoods UML



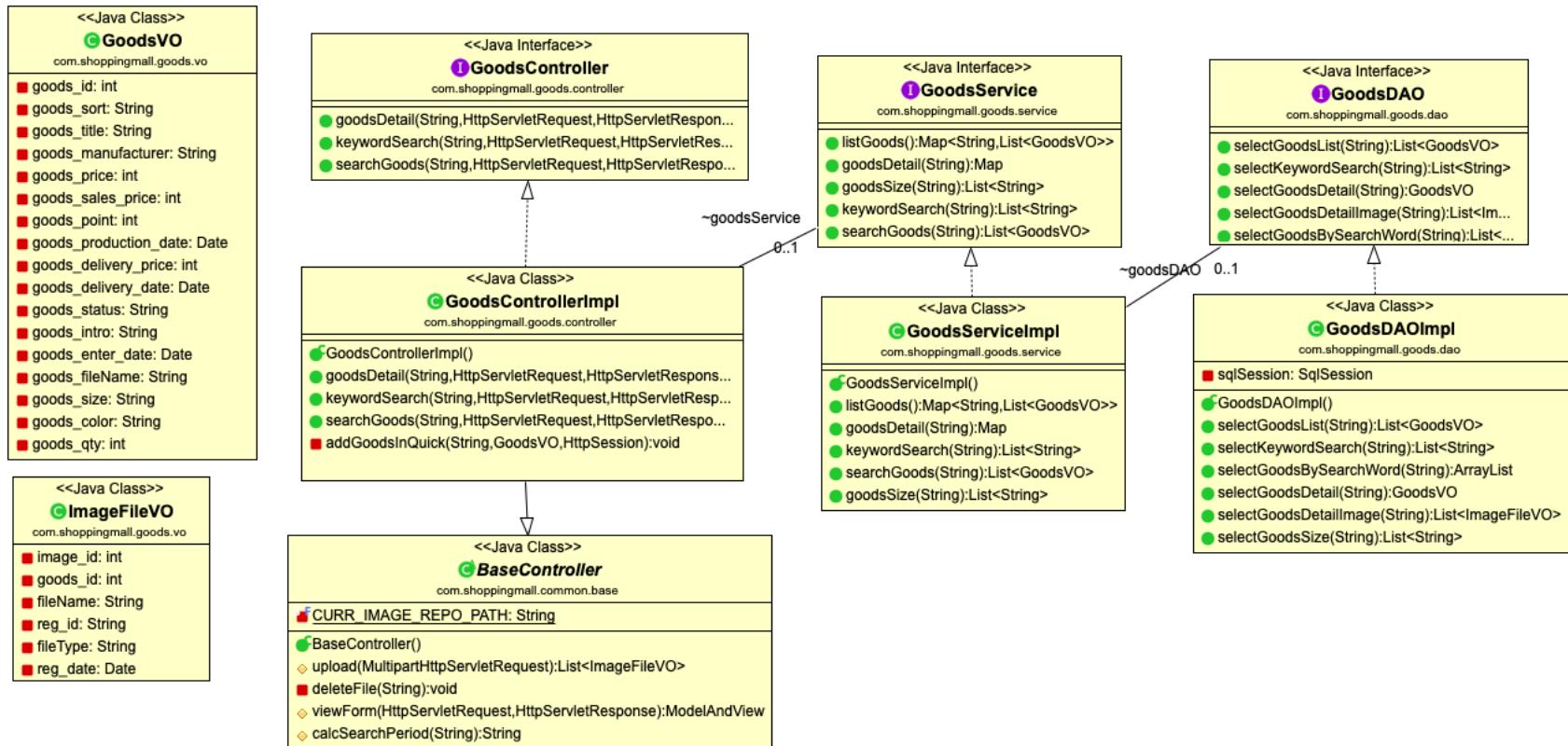
adminMember UML



adminOrder UML



Cart UML



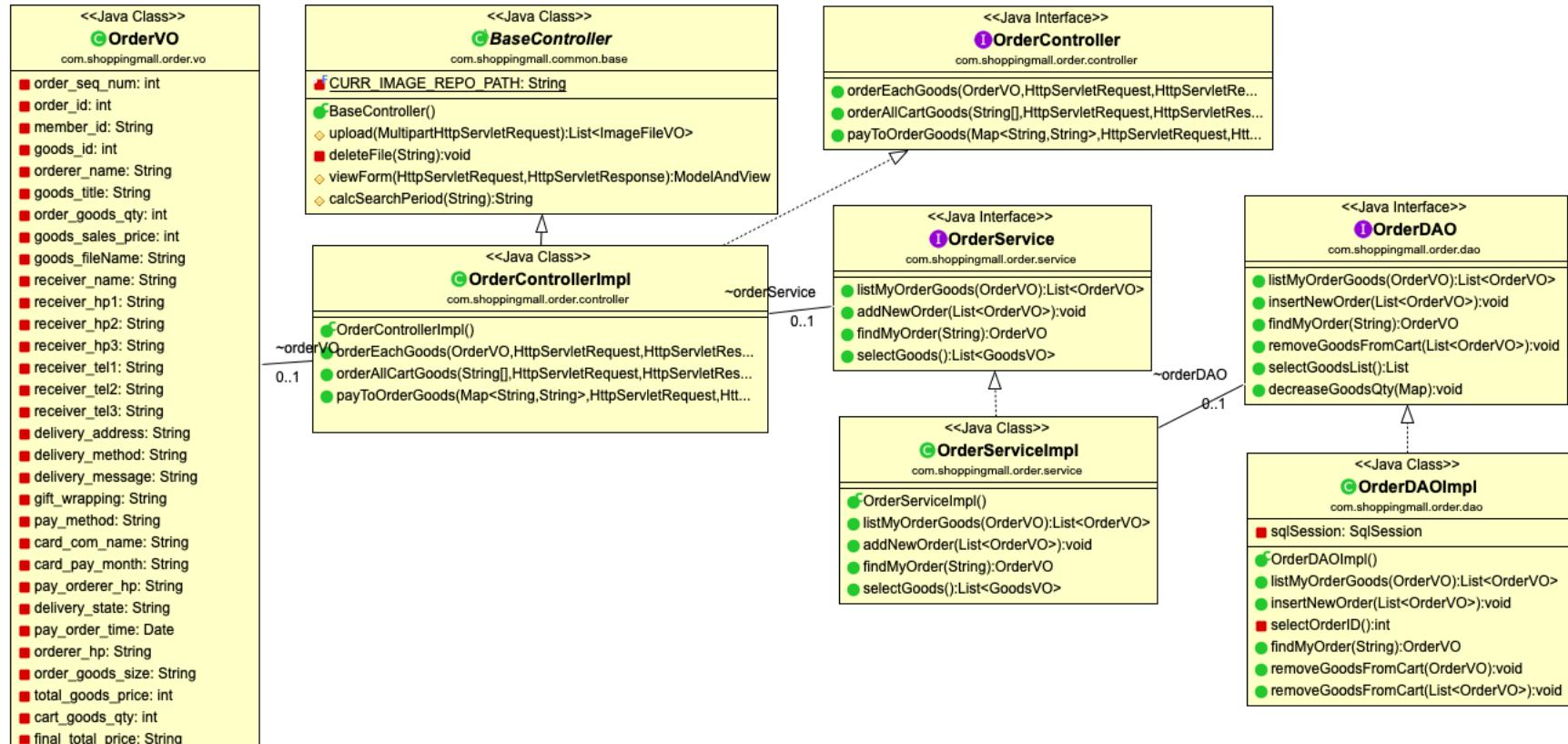
Goods UML



member UML



MyPage UML



Order UML

# Index



01

서론

프로젝트 목적

프로젝트 개요

02

본론

프로젝트 구성도

프로젝트 주요 기술

03

결론

개선 사항

프로젝트 후기

- 데이터베이스 종류 : Oracle
- 사용 버전 : 11g
- 사용 툴 : SqlDeveloper

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 GOODS_ID	NUMBER(20,0)	No	(null)	1	상품번호
2 GOODS_SORT	VARCHAR2(50 BYTE)	Yes	(null)	2	상품종류
3 GOODS_TITLE	VARCHAR2(100 BYTE)	Yes	(null)	3	상품이름
4 GOODS_MANUFACTURER	VARCHAR2(50 BYTE)	Yes	(null)	4	제조사
5 GOODS_PRICE	NUMBER(10,0)	Yes	(null)	5	상품정가
6 GOODS_SALES_PRICE	NUMBER(10,0)	Yes	(null)	6	상품판매가격
7 GOODS_POINT	NUMBER(10,0)	Yes	(null)	7	상품포인트
8 GOODS_PRODUCTION_DATE	DATE	Yes	(null)	8	제조날짜
9 GOODS_DELIVERY_PRICE	NUMBER(10,0)	Yes	(null)	9	배송비
10 GOODS_DELIVERY_DATE	DATE	Yes	(null)	10	상품배송일
11 GOODS_STATUS	VARCHAR2(50 BYTE)	Yes	(null)	11	상품분류
12 GOODS_INTRO	VARCHAR2(2000 BYTE)	Yes	(null)	12	상품소개
13 GOODS_ENTER_DATE	DATE	Yes	sysdate	13	상품입고일

상품의 정보를 저장하는 테이블

```
CREATE TABLE "SOCCER"."GOODS"
("GOODS_ID" NUMBER(20,0),
 "GOODS_SORT" VARCHAR2(50 BYTE),
 "GOODS_TITLE" VARCHAR2(100 BYTE),
 "GOODS_MANUFACTURER" VARCHAR2(50 BYTE),
 "GOODS_PRICE" NUMBER(10,0),
 "GOODS_SALES_PRICE" NUMBER(10,0),
 "GOODS_POINT" NUMBER(10,0),
 "GOODS_PRODUCTION_DATE" DATE,
 "GOODS_DELIVERY_PRICE" NUMBER(10,0),
 "GOODS_DELIVERY_DATE" DATE,
 "GOODS_STATUS" VARCHAR2(50 BYTE),
 "GOODS_INTRO" VARCHAR2(2000 BYTE),
 "GOODS_ENTER_DATE" DATE DEFAULT sysdate,
 PRIMARY KEY ("GOODS_ID"))
```

GOODS 테이블 생성 SQL

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	GOODS_ID	NUMBER(20,0)	No	(null)	1	상품번호
2	GOODS_SIZE	VARCHAR2(50 BYTE)	No	(null)	2	상품사이즈
3	GOODS_COLOR	VARCHAR2(100 BYTE)	No	(null)	3	상품색상
4	GOODS_QTY	NUMBER(20,0)	Yes	(null)	4	재고수량

```
CREATE TABLE "SOCCER"."GOODS_DETAIL"
("GOODS_ID" NUMBER(20,0),
"GOODS_SIZE" VARCHAR2(50 BYTE),
"GOODS_COLOR" VARCHAR2(100 BYTE) NOT NULL ENABLE,
"GOODS_QTY" NUMBER(20,0),
CONSTRAINT "GOODS_DETAIL_PK" PRIMARY KEY ("GOODS_ID", "GOODS_SIZE", "GOODS_COLOR"))
```

- 상품의 사이즈, 색상, 재고 수량을 저장하는 테이블
- GOODS 테이블과 조인하여 사용

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 IMAGE_ID	NUMBER(20,0)	No	(null)	1	0 이미지번호
2 GOODS_ID	NUMBER(20,0)	Yes	(null)	2	상품번호
3 FILENAME	VARCHAR2(250 BYTE)	Yes	(null)	3	0 이미지파일명
4 REG_ID	VARCHAR2(20 BYTE)	Yes	(null)	4	등록자아이디
5 FILETYPE	VARCHAR2(40 BYTE)	Yes	(null)	5	0 이미지파일종류
6 REG_DATE	DATE	Yes	sysdate	6	등록일

```

CREATE TABLE "SOCCER"."GOODS_IMAGE"
("IMAGE_ID" NUMBER(20,0),
"GOODS_ID" NUMBER(20,0),
"FILENAME" VARCHAR2(250 BYTE),
"REG_ID" VARCHAR2(20 BYTE),
"FILETYPE" VARCHAR2(40 BYTE),
"REG_DATE" DATE DEFAULT sysdate,
PRIMARY KEY ("IMAGE_ID"))

```

- 상품의 이미지를 업로드한 정보를 담는 테이블

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 MEMBER_ID	VARCHAR2(20 BYTE)	No	(null)	1	회원아이디
2 MEMBER_PW	VARCHAR2(30 BYTE)	Yes	(null)	2	비밀번호
3 MEMBER_NAME	VARCHAR2(50 BYTE)	Yes	(null)	3	회원이름
4 MEMBER_GENDER	VARCHAR2(10 BYTE)	Yes	(null)	4	성별
5 TEL1	VARCHAR2(20 BYTE)	Yes	(null)	5	유선전화번호1
6 TEL2	VARCHAR2(20 BYTE)	Yes	(null)	6	유선전화번호2
7 TEL3	VARCHAR2(20 BYTE)	Yes	(null)	7	유선전화번호3
8 HP1	VARCHAR2(20 BYTE)	Yes	(null)	8	휴대폰번호1
9 HP2	VARCHAR2(20 BYTE)	Yes	(null)	9	휴대폰번호2
10 HP3	VARCHAR2(20 BYTE)	Yes	(null)	10	휴대폰번호3
11 SMS_YN	VARCHAR2(20 BYTE)	Yes	(null)	11	SMS메시지수신여부
12 EMAIL1	VARCHAR2(20 BYTE)	Yes	(null)	12	이메일주소1
13 EMAIL2	VARCHAR2(20 BYTE)	Yes	(null)	13	이메일주소2
14 EMAIL_YN	VARCHAR2(20 BYTE)	Yes	(null)	14	이메일수신여부
15 ZIPCODE	VARCHAR2(20 BYTE)	Yes	(null)	15	우편번호
16 ROAD_ADDRESS	VARCHAR2(500 BYTE)	Yes	(null)	16	도로명주소
17 JIBUN_ADDRESS	VARCHAR2(500 BYTE)	Yes	(null)	17	지번주소
18 DETAIL_ADDRESS	VARCHAR2(500 BYTE)	Yes	(null)	18	나머지주소
19 BIRTH_YEAR	VARCHAR2(20 BYTE)	Yes	(null)	19	생년월일년
20 BIRTH_MONTH	VARCHAR2(20 BYTE)	Yes	(null)	20	생년월일월
21 BIRTH_DAY	VARCHAR2(20 BYTE)	Yes	(null)	21	생년월일일
22 JOIN_DATE	DATE	Yes	sysdate	22	회원가입일자
23 DELETE_YN	VARCHAR2(20 BYTE)	Yes	'N'	23	회원탈퇴유무

```

CREATE TABLE "SOCCER"."MEMBER"
(
    "MEMBER_ID" VARCHAR2(20 BYTE),
    "MEMBER_PW" VARCHAR2(30 BYTE),
    "MEMBER_NAME" VARCHAR2(50 BYTE),
    "MEMBER_GENDER" VARCHAR2(10 BYTE),
    "TEL1" VARCHAR2(20 BYTE),
    "TEL2" VARCHAR2(20 BYTE),
    "TEL3" VARCHAR2(20 BYTE),
    "HP1" VARCHAR2(20 BYTE),
    "HP2" VARCHAR2(20 BYTE),
    "HP3" VARCHAR2(20 BYTE),
    "SMS_YN" VARCHAR2(20 BYTE),
    "EMAIL1" VARCHAR2(20 BYTE),
    "EMAIL2" VARCHAR2(20 BYTE),
    "EMAIL_YN" VARCHAR2(20 BYTE),
    "ZIPCODE" VARCHAR2(20 BYTE),
    "ROAD_ADDRESS" VARCHAR2(500 BYTE),
    "JIBUN_ADDRESS" VARCHAR2(500 BYTE),
    "DETAIL_ADDRESS" VARCHAR2(500 BYTE),
    "BIRTH_YEAR" VARCHAR2(20 BYTE),
    "BIRTH_MONTH" VARCHAR2(20 BYTE),
    "BIRTH_DAY" VARCHAR2(20 BYTE),
    "JOIN_DATE" DATE DEFAULT sysdate,
    "DELETE_YN" VARCHAR2(20 BYTE) DEFAULT 'N',
    PRIMARY KEY ("MEMBER_ID")
)

```

회원 정보를 담는 테이블

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 CART_ID	NUMBER(10,0)	No	(null)	1	장바구니번호
2 GOODS_ID	NUMBER(20,0)	Yes	(null)	2	상품번호
3 MEMBER_ID	VARCHAR2(20 BYTE)	Yes	(null)	3	회원아이디
4 DELETE_YN	VARCHAR2(20 BYTE)	Yes	'N'	4	장바구니삭제여부
5 REG_DATE	DATE	Yes	sysdate	5	생성일자
6 CART_GOODS_QTY	NUMBER(4,0)	Yes	1	6	상품수량
7 CART_GOODS_SIZE	VARCHAR2(20 BYTE)	Yes	(null)	7	제품사이즈

```
CREATE TABLE "SOCCER"."CART"
("CART_ID" NUMBER(10,0),
"GOODS_ID" NUMBER(20,0),
"MEMBER_ID" VARCHAR2(20 BYTE),
"DELETE_YN" VARCHAR2(20 BYTE) DEFAULT 'N',
"REG_DATE" DATE DEFAULT sysdate,
"CART_GOODS_QTY" NUMBER(4,0) DEFAULT 1,
"CART_GOODS_SIZE" VARCHAR2(20 BYTE),
PRIMARY KEY ("CART_ID"))
```

장바구니 정보를 담는 테이블

# DataBase 설계(6)

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ORDER_SEQ_NUM	NUMBER(20,0)	No	(null)		1 주문상품일련번호
2	ORDER_ID	NUMBER(20,0)	Yes	(null)		2 주문번호
3	MEMBER_ID	VARCHAR2(20 BYTE)	Yes	(null)		3 주문자아이디
4	GOODS_ID	NUMBER(20,0)	Yes	(null)		4 상품이름
5	ORDERER_NAME	VARCHAR2(50 BYTE)	Yes	(null)		5 주문수량
6	GOODS_TITLE	VARCHAR2(100 BYTE)	Yes	(null)		6 상품이름
7	ORDER_GOODS_QTY	NUMBER(20,0)	Yes	(null)		7 주문수량
8	GOODS_SALES_PRICE	NUMBER(20,0)	Yes	(null)		8 상품판매가격
9	GOODS_FILENAME	VARCHAR2(60 BYTE)	Yes	(null)		9 상품이미지파일이름
10	RECEIVER_NAME	VARCHAR2(50 BYTE)	Yes	(null)		10 수령자이름
11	RECEIVER_HP1	VARCHAR2(20 BYTE)	Yes	(null)		11 수령자휴대폰번호1
12	RECEIVER_HP2	VARCHAR2(20 BYTE)	Yes	(null)		12 수령자휴대폰번호2
13	RECEIVER_HP3	VARCHAR2(20 BYTE)	Yes	(null)		13 수령자휴대폰번호3
14	RECEIVER_TEL1	VARCHAR2(20 BYTE)	Yes	(null)		14 수령자유선번호1
15	RECEIVER_TEL2	VARCHAR2(20 BYTE)	Yes	(null)		15 수령자유선번호2
16	RECEIVER_TEL3	VARCHAR2(20 BYTE)	Yes	(null)		16 수령자유선번호3
17	DELIVERY_ADDRESS	VARCHAR2(500 BYTE)	Yes	(null)		17 배송주소
18	DELIVERY_METHOD	VARCHAR2(40 BYTE)	Yes	(null)		18 배송방법
19	DELIVERY_MESSAGE	VARCHAR2(300 BYTE)	Yes	(null)		19 부재 시 전달 메시지
20	GIFT_WRAPPING	VARCHAR2(20 BYTE)	Yes	(null)		20 상품포장여부
21	PAY_METHOD	VARCHAR2(200 BYTE)	Yes	(null)		21 결제방법
22	CARD_COM_NAME	VARCHAR2(50 BYTE)	Yes	(null)		22 결제카드회사이름
23	CARD_PAY_MONTH	VARCHAR2(20 BYTE)	Yes	(null)		23 할부개월수
24	PAY_ORDERER_HP	VARCHAR2(20 BYTE)	Yes	(null)		24 휴대폰결제번호
25	DELIVERY_STATE	VARCHAR2(20 BYTE)	Yes	'delivery_prepared'		25 배송상태
26	PAY_ORDER_TIME	DATE	Yes	sysdate		26 결제시간
27	ORDERER_HP	VARCHAR2(50 BYTE)	Yes	(null)		27 주문자휴대폰번호
28	ORDER_GOODS_SIZE	VARCHAR2(20 BYTE)	Yes	(null)		28 주문상품사이즈

주문 정보를 담는 테이블

```

CREATE TABLE "SOCCER"."ORDERS"
(
    "ORDER_SEQ_NUM" NUMBER(20,0),
    "ORDER_ID" NUMBER(20,0),
    "MEMBER_ID" VARCHAR2(20 BYTE),
    "GOODS_ID" NUMBER(20,0),
    "ORDERER_NAME" VARCHAR2(50 BYTE),
    "GOODS_TITLE" VARCHAR2(100 BYTE),
    "ORDER_GOODS_QTY" NUMBER(20,0),
    "GOODS_SALES_PRICE" NUMBER(20,0),
    "GOODS_FILENAME" VARCHAR2(60 BYTE),
    "RECEIVER_NAME" VARCHAR2(50 BYTE),
    "RECEIVER_HP1" VARCHAR2(20 BYTE),
    "RECEIVER_HP2" VARCHAR2(20 BYTE),
    "RECEIVER_HP3" VARCHAR2(20 BYTE),
    "RECEIVER_TEL1" VARCHAR2(20 BYTE),
    "RECEIVER_TEL2" VARCHAR2(20 BYTE),
    "RECEIVER_TEL3" VARCHAR2(20 BYTE),
    "DELIVERY_ADDRESS" VARCHAR2(500 BYTE),
    "DELIVERY_METHOD" VARCHAR2(40 BYTE),
    "DELIVERY_MESSAGE" VARCHAR2(300 BYTE),
    "GIFT_WRAPPING" VARCHAR2(20 BYTE),
    "PAY_METHOD" VARCHAR2(200 BYTE),
    "CARD_COM_NAME" VARCHAR2(50 BYTE),
    "CARD_PAY_MONTH" VARCHAR2(20 BYTE),
    "PAY_ORDERER_HP" VARCHAR2(20 BYTE),
    "DELIVERY_STATE" VARCHAR2(20 BYTE) DEFAULT 'delivery_prepared',
    "PAY_ORDER_TIME" DATE DEFAULT sysdate,
    "ORDERER_HP" VARCHAR2(50 BYTE),
    "ORDER_GOODS_SIZE" VARCHAR2(20 BYTE),
    PRIMARY KEY ("ORDER_SEQ_NUM")
)

```

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://java.sun.com/xml/
3   <filter>
4     <filter-name>encodingFilter</filter-name>
5     <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
6     <init-param>
7       <param-name>encoding</param-name>
8       <param-value>UTF-8</param-value>
9     </init-param>
10    </filter>
11    <filter-mapping>
12      <filter-name>encodingFilter</filter-name>
13      <url-pattern>/*</url-pattern>
14    </filter-mapping>
15    <context-param>
16      <param-name>contextConfigLocation</param-name>
17      <param-value>/WEB-INF/spring/root-context.xml</param-value>
18    </context-param>
19    <listener>
20      <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
21    </listener>
22    <servlet>
23      <servlet-name>appServlet</servlet-name>
24      <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
25      <init-param>
26        <param-name>contextConfigLocation</param-name>
27        <param-value>/WEB-INF/spring/appServlet/servlet-context.xml</param-value>
28      </init-param>
29      <load-on-startup>1</load-on-startup>
30    </servlet>
31    <filter-mapping>
32      <servlet-name>appServlet</servlet-name>
33      <url-pattern>/</url-pattern>
34    </filter-mapping>
35  </web-app>

```

- ContextLoaderListner : WebServelt컨테이너 생성 후 스프링에서 DispatcherServlet을 생성하하기 전 applicationContext2.xml파일을 로딩하여 스프링 컨테이너 구동 : Service 구현 객체나 DAO 객체 메모리 생성
- DispatcherServlet : \*.do의 요청이 들어 오면 두번째 스프링 컨테이너를 생성 하며 presentation-layer.xml 설정 되어 있는 bean객체들을 생성 : controller 객체를 메모리에 생성
- encodingfilter : controller로 들어오는 모든 /\*요청에 대해서 UTF-8 encoding으로 처리하기 위한 설정

```
39      <!-- mybatis-spring -->
40      <dependency>
41          <groupId>org.mybatis</groupId>
42          <artifactId>mybatis-spring</artifactId>
43          <version>2.0.1</version>
44      </dependency>
45
46
47      <!-- mybatis -->
48      <dependency>
49          <groupId>org.mybatis</groupId>
50          <artifactId>mybatis</artifactId>
51          <version>3.5.1</version>
52      </dependency>
53
54      <!-- json -->
55      <dependency>
56          <groupId>net.sf.json-lib</groupId>
57          <artifactId>json-lib</artifactId>
58          <version>2.4</version>
59          <classifier>jdk15</classifier>
60      </dependency>
61
62      <!-- jquery -->
63      <dependency>
64          <groupId>org.webjars</groupId>
65          <artifactId>jquery</artifactId>
66          <version>3.4.1</version>
67      </dependency>
```

- mybatis, mybatis-spring : mybatis와 스프링을 연동 하기 위한 API
- net.sf.json-lib : 스프링에서 json데이터를 사용하기 위한 API
- jquery : 스프링에서 jquery를 사용하기 위한 API

# 설정 파일 – pom.xml(2)

```
16<dependencies>
17    <!-- tiles 설정 (core, jsp, servlet) -->
18    <!-- tiles-core -->
19<dependency>
20    <groupId>org.apache.tiles</groupId>
21    <artifactId>tiles-core</artifactId>
22    <version>3.0.8</version>
23</dependency>
24
25    <!-- tiles-jsp -->
26<dependency>
27    <groupId>org.apache.tiles</groupId>
28    <artifactId>tiles-jsp</artifactId>
29    <version>3.0.8</version>
30</dependency>
31
32    <!-- tiles-servlet -->
33<dependency>
34    <groupId>org.apache.tiles</groupId>
35    <artifactId>tiles-servlet</artifactId>
36    <version>3.0.8</version>
37</dependency>
```

tiles-core, tiles-jsp, tiles: tiles를 사용하기 위한 API

```
69      <!-- net.coobird/thumbnailator -->
70      <dependency>
71          <groupId>net.coobird</groupId>
72          <artifactId>thumbnailator</artifactId>
73          <version>0.4.8</version>
74      </dependency>
75
76
77      <!-- file upload -->
78      <dependency>
79          <groupId>commons-fileupload</groupId>
80          <artifactId>commons-fileupload</artifactId>
81          <version>1.4</version>
82      </dependency>
83
84      <!-- spring-jdbc -->
85      <dependency>
86          <groupId>org.springframework</groupId>
87          <artifactId>spring-jdbc</artifactId>
88          <version>${org.springframework-version}</version>
89      </dependency>
```

- thumbnailator: 서버에서 썸네일 이미지 생성하기 위한 API
- commons-fileupload : 파일 업로드 API
- spring-jdbc : Spring에서 jdbc를 사용하기 위한 API

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xmlns:aop="http://www.springframework.org/schema/aop"
5   xmlns:tx="http://www.springframework.org/schema/tx"
6   xsi:schemaLocation="http://www.springframework.org/schema/beans https://www.springframework.org/schema/beans/spring-beans.xsd
7     http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.3.xsd
8     http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-4.3.xsd
9     http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-4.3.xsd">
10
11 <!-- Root Context: defines shared resources visible to all other web components -->
12 <aop:aspectj-autoproxy/>
13 <!-- JDBC Properties 설정 -->
14 <bean id="propertyPlaceholderConfigurer" class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
15   <property name="locations">
16     <value>/WEB-INF/config/jdbc/jdbc.properties</value> jdbc properties 경로 설정
17   </property>
18 </bean>
19 <!-- DataSource 설정 --> jdbc.properties에 매팅
20 <bean id="dataSource" class="org.apache.ibatis.datasource.pooled.PooledDataSource">
21   <property name="driver" value="${jdbc.driverClassName}"/>
22   <property name="url" value="${jdbc.url}"/>
23   <property name="username" value="${jdbc.username}"/>
24   <property name="password" value="${jdbc.password}"/>
25 </bean>
```

```
<!-- <tx:annotation-driver transaction-manager="DataSourceTransactionManager"/>
<!-- Transaction 설정 --> 어노테이션으로 설정한 트랜잭션 처리
<bean id="transactionManager" class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
    <property name="dataSource" ref="dataSource"></property>
</bean>

<!-- Spring과 Mybatis 연동 설정 -->
<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
    <property name="dataSource" ref="dataSource"/>
    <property name="configLocation" value="classpath:mybatis/models/modelConfig.xml"/>
    <property name="mapperLocations" value="classpath:mybatis/mappers/*.xml"/>
</bean>
<!-- SqlSessionTemplate Constructor주입(생성자 주입) Mybatis를 사용하기 위한 sqlsessionfactory 설정
<bean id="sqlSession" class="org.mybatis.spring.SqlSessionTemplate">
    <constructor-arg ref="sqlSessionFactory"></constructor-arg>
</bean>
</beans>
```

```

13      <!-- Enables the Spring M
14      <mvc:annotation-driven /> 어노테이션 선언한 클래스들을 읽어내기 위한 설정
15
16      <!-- Handles HTTP GET requests for /resources/** by efficiently serving
17          up static resources in the ${webappRoot}/resources directory -->
18      <mvc:resources mapping="/resources/**" location="/resources/" />
19
20      <!-- Resolves views selected
21          in the /WEB-INF/views directory --> 공통적으로 사용하는 고정된 파일들 경로 설정
22
23      <!-- 태일즈 설정 --> 태일즈를 사용하기 위한 설정
24      <beans:bean id="tilesConfigurer"
25          class="org.springframework.web.servlet.view.tiles3.TilesConfigurer">
26          <beans:property name="definitions">
27              <beans:list>
28                  <beans:value>classpath:.tiles/*.xml</beans:value>
29              </beans:list> 태일즈 파일 경로
30          </beans:property>
31          <beans:property name="preparerFactoryClass"
32              value="org.springframework.web.servlet.view.tiles3.SpringBeanPreparerFactory" />
33      </beans:bean>
34      <context:component-scan base-package="com.shoppingmall" />
35      <beans:bean id="viewResolver"
36          class="org.springframework.web.servlet.view.UrlBasedViewResolver">
37          <beans:property name="viewClass">
38              value="org.springframework.web.servlet.view.tiles3.TilesView" />
39      </beans:bean>

```

```
40  
41      <!-- 2단계 3단계 요청에 대한 인터셉터 설정 -->  
42      <mvc:interceptors>    인터셉터가 읽어 드릴 url요청 패턴 설정  
43          <mvc:interceptor>  
44              <mvc:mapping path="/*/*.do" />  
45              <mvc:mapping path="/*/*/*.do" />  
46              <beans:bean  
47                  class="com.shoppingmall.common.interceptor.ViewNameInterceptor" />  
48          </mvc:interceptor>  
49      </mvc:interceptors>  
50  
51      <!-- Multipart ViewResolver -->    스프링에서 제공하는 파일업로드 설정  
52      <beans:bean id="multipartResolver"  
53          class="org.springframework.web.multipart.commons.CommonsMultipartResolver">  
54          <beans:property name="maxUploadSize" value="52428800" />  
55          <beans:property name="maxInMemorySize" value="1000000" />  
56          <beans:property name="defaultEncoding" value="utf-8" />  
57      </beans:bean>  
58  
59  </beans:beans>
```

# VO(Value Object)

```
public class GoodsVO {  
    private int goods_id;  
    private String goods_sort;  
    private String goods_title;  
    private String goods_manufacturer;  
    private int goods_price;  
    private int goods_sales_price;  
    private int goods_point;  
    private Date goods_production_date;  
    private int goods_delivery_price;  
    private Date goods_delivery_date;  
    private String goods_status;  
    private String goods_intro;  
    private Date goods_enter_date;  
    private String goods_fileName;  
    private String goods_size;  
    private String goods_color;  
    private int goods_qty;
```

```
5 public class ImageFileVO {  
6     private int image_id;  
7     private int goods_id;  
8     private String fileName;  
9     private String reg_id;  
10    private String fileType;  
11    private Date reg_date;  
12}
```

- 상품 정보 데이터를 처리 시 사용하는 VO(Value Object)
- GoodsVO : 상품 정보
- ImageFileVO : 업로드 이미지의 정보

```

@Controller("mainController")
@EnableAspectJAutoProxy
public class MainController extends BaseController{

    @Autowired
    private GoodsService goodsService;

    //메인화면에서 전체 상품분류 별로 나누어 보여줌(인기상품, 새로운 상품 출시, 스테디셀러)
    @RequestMapping(value="/main/main.do", method= {RequestMethod.POST, RequestMethod.GET})
    public ModelAndView main(HttpServletRequest request, HttpServletResponse response) throws Exception {
        System.out.println("controller1");
        HttpSession session;
        ModelAndView mav = new ModelAndView();
        String viewName= (String)request.getAttribute("viewName");
        mav.setViewName(viewName);
        System.out.println("controller2");
        session = request.getSession();
        session.setAttribute("side_menu", "user");
        System.out.println("controller3");
        Map<String, List<GoodsVO>>goodsMap = goodsService.listGoods();
        mav.addObject("goodsMap", goodsMap);
        System.out.println(mav);
        return mav;
    }
}

```

- 메인 화면 요청시 작동하는 controller
- @Controller : applicationContext에서 component-scan에서 읽어 스캔 하기 위한 어노테이션 (Annotation)
- Autowired : GoodsService 클래스를 DI(Dependency Injection)를 하기 위한 어노테이션

# GoodsControllerImpl(1)

```

@Controller("goodsController")
@RequestMapping(value="/goods")
public class GoodsControllerImpl extends BaseController implements GoodsController {
    @Autowired
    GoodsService goodsService;

    // 상품 상세보기를 위한 요청 처리
    @RequestMapping(value="/goodsDetail.do" ,method = RequestMethod.GET)
    public ModelAndView goodsDetail(@RequestParam("goods_id") String goods_id,
                                    HttpServletRequest request,
                                    HttpServletResponse response) throws Exception {
        String viewName=(String)request.getAttribute("viewName");
        HttpSession session=request.getSession();
        Map goodsMap=goodsService.goodsDetail(goods_id);
        Map<String, List> goodsSizeMap = new HashMap<String, List>();
        goodsSizeMap.put("goodsSize", goodsService.goodsSize(goods_id));

        ModelAndView mav = new ModelAndView(viewName);
        mav.addObject("goodsMap", goodsMap);
        mav.addObject("goodsSizeMap", goodsSizeMap);
        GoodsVO goodsVO=(GoodsVO)goodsMap.get("goodsVO");
        System.out.println(goodsVO.toString());
        addGoodsInQuick(goods_id,goodsVO,session); • 상품 상세 보기 시 quickMenu 리스트에 추가
        return mav;
    }

    //검색창에 검색어 입력시 검색창에 검색어에 대한 트리거제안 목록 출력
    @RequestMapping(value="/keywordSearch.do",method = RequestMethod.GET,produces = "application/json")
    public @ResponseBody String keywordSearch(@RequestParam("keyWord") String keyWord)
    {
        return null;
    }
}

```

메인 페이지에서 상품을 클릭(/goods/goodsDetail.do 요청)하였을 때 동작하는 controller

# GoodsControllerImpl(2)

```
//검색창에 검색어 입력시 검색창에 검색어에 대한 트리거제안 목록 출력
@RequestMapping(value="/keywordSearch.do",method = RequestMethod.GET,produces = "application/text; charset=utf8")
public @ResponseBody String keywordSearch(@RequestParam("keyword") String keyword,
                                         HttpServletRequest request,
                                         HttpServletResponse response) throws Exception{
    response.setContentType("text/html;charset=utf-8");
    response.setCharacterEncoding("utf-8");
    if(keyword == null || keyword.equals(""))
        return null ;

    keyword = keyword.toUpperCase();
    List<String> keywordList =goodsService.keywordSearch(keyword);

    JSONObject json0bject = new JSONObject();
    json0bject.put("keyword", keywordList);

    String jsonInfo = json0bject.toString();
    return jsonInfo ;
}

@RequestMapping(value="/searchGoods.do" ,method = RequestMethod.GET)
public ModelAndView searchGoods(@RequestParam("searchWord") String searchWord,
                                HttpServletRequest request,
                                HttpServletResponse response) throws Exception{
    String viewName=(String)request.getAttribute("viewName");
    List<GoodsVO> goodsList=goodsService.searchGoods(searchWord);
    ModelAndView mav = new ModelAndView(viewName);
    mav.addObject("goodsList", goodsList);
    return mav;
}
```

- `ResponseBody` : JSON형식으로 만든 데이터를 화면 단에 데이터를 보내주기 위한 어노테이션
- `/keywordSearch.dp` : 검색창에 입력한 단어에 따라 검색 어 트리거 제안
- `/searchGoods.do` : 검색 버튼 클릭 시 요청 받는 controller

# GoodsControllerImpl(3)

```

private void addGoodsInQuick(String goods_id, GoodsVO goodsVO, HttpSession session){
    boolean already_existed=false;
    List<GoodsVO> quickGoodsList;
    quickGoodsList=(ArrayList<GoodsVO>)session.getAttribute("quickGoodsList");

    if(quickGoodsList!=null){
        if(quickGoodsList.size() < 4){
            for(int i=0; i<quickGoodsList.size(); i++){
                GoodsVO _goodsBean=(GoodsVO)quickGoodsList.get(i);
                if(goods_id.equals(_goodsBean.getGoods_id())){
                    already_existed=true;
                    break;
                }
            }
            if(already_existed==false){
                quickGoodsList.add(goodsVO);
            }
        }
    } else{
        quickGoodsList =new ArrayList<GoodsVO>;
        quickGoodsList.add(goodsVO);

    }
    session.setAttribute("quickGoodsList",quickGoodsList);
    session.setAttribute("quickGoodsListNum", quickGoodsList.size());
}

@RequestMapping(value="/goodsList.do", method= {RequestMethod.POST, RequestMethod.GET})
public ModelAndView byTypeGoods(@RequestParam String goods_sort, HttpServletRequest request,
                                HttpServletResponse response) throws Exception {
    HttpSession session;
    ModelAndView mav = new ModelAndView();
    String viewName= (String)request.getAttribute("viewName");
    mav.setViewName(viewName);
    session = request.getSession();
    session.setAttribute("side_menu", "user");
    System.out.println("controller3");
    List<GoodsVO> goodsList = goodsService.byTypeGoods(goods_sort);
    mav.addObject("goods_sort", goods_sort);
    mav.addObject("goodsList", goodsList);
    return mav;
}

```

- goodsDetail에서 호출
- 이미 리스트에 있으면 break로 빠져나옴
- 세션에 저장 후 로그 아웃 시 제거

- addGoodsInQuick() : GoodsDetail에서 호출하는 메소드(퀵메뉴)
- /goodsList.do : 상품 종목별로 리스트페이지 클릭 시 호출

# GoodsServiceImp(1)

```

@Service("goodsService")
@Transactional(propagation=Propagation.REQUIRED)
public class GoodsServiceImpl implements GoodsService{

    @Autowired
    GoodsDAO goodsDAO;

    //메인화면 처리를 위한 비즈니스영역
    @Override
    public Map<String, List<GoodsVO>> listGoods() throws Exception {
        Map<String, List<GoodsVO>> goodsMap = new HashMap<String, List<GoodsVO>>();
        List<GoodsVO> goodsList = goodsDAO.selectGoodsList("bestseller");
        goodsMap.put("bestseller", goodsList);

        goodsList = goodsDAO.selectGoodsList("newproduct");
        goodsMap.put("newproduct", goodsList);

        goodsList = goodsDAO.selectGoodsList("steadyseller");
        goodsMap.put("steadyseller", goodsList);

        return goodsMap;
    }

    //상품 클릭시 상세 정보를 처리하는 메서드
    @Override
    public Map goodsDetail(String _goods_id) throws Exception {
        Map goodsMap = new HashMap();
        GoodsVO goodsVO = goodsDAO.selectGoodsDetail(_goods_id);
        goodsMap.put("goodsVO", goodsVO);

        List<ImageFileVO> imagelist = goodsDAO.selectGoodsDetailImage(_goods_id);
        goodsMap.put("imagelist", imagelist);
        return goodsMap;
    }

    //검색어 카테고리 목록을 처리하는 메서드
    @Override
    public List<String> keywordSearch(String keyword) throws Exception {
        List<String> list = goodsDAO.selectKeywordSearch(keyword);
        return list;
    }
}

```

```

58  @Override
59  public List<GoodsVO> searchGoods(String searchWord) throws Exception {
60      List goodsList = goodsDAO.selectGoodsBySearchWord(searchWord);
61      return goodsList;
62  }
63
64  @Override
65  public List<String> goodsSize(String _goods_id) throws Exception {
66      List goodsSize = goodsDAO.selectGoodsSize(_goods_id);
67      return goodsSize;
68  }
69
70  @Override
71  public List<GoodsVO> byTypeGoods(String goods_sort) throws Exception {
72      List<GoodsVO> goodsList = goodsDAO.selectBySort(goods_sort);
73      return goodsList;
74  }
75
76 }

```

- @Service : @Component 대신 비즈니스 로직에서 사용하는 어노테이션
- GoodsController에서 호출하는 비즈니스 로직

# GoodsDAOImpl(1)

```
16 @Repository("goodsDAO")
17 public class GoodsDAOImpl implements GoodsDAO{
18@     @Autowired
19     private SqlSession sqlSession;
20
21@     @Override//상품 전체 목록
22     public List<GoodsVO> selectGoodsList(String goodsStatus ) throws DataAccessException {
23         System.out.println("dao1");
24         List<GoodsVO> goodsList=(ArrayList)sqlSession.selectList("mapper.goods.selectGoodsList",goodsStatus);
25         System.out.println("dao2");
26         return goodsList;
27     }
28@     @Override// 상품 이름 입력시 가져오는 상품이름목록
29     public List<String> selectKeywordSearch(String keyword) throws DataAccessException {
30         List<String> list=(ArrayList)sqlSession.selectList("mapper.goods.selectKeywordSearch",keyword);
31         return list;
32     }
33
34@     @Override// 상품 검색 후 가져오는 상품 목록
35     public ArrayList selectGoodsBySearchWord(String searchWord) throws DataAccessException{
36         ArrayList list=(ArrayList)sqlSession.selectList("mapper.goods.selectGoodsBySearchWord",searchWord);
37         return list;
38     }
39
40@     @Override//상품 상세보기 페이지에서 가져오는 목록
41     public GoodsVO selectGoodsDetail(String goods_id) throws DataAccessException{
42         GoodsVO goodsVO=(GoodsVO)sqlSession.selectOne("mapper.goods.selectGoodsDetail",goods_id);
43         return goodsVO;
44     }
```

DB(DataBase)를 처리하는 persistence 영역 로직

# GoodsDAOImpl(2)

```
46@ Override//상품 상세보기 페이지에서 사용하는 이미지 목록
47public List<ImageFileVO> selectGoodsDetailImage(String goods_id) throws DataAccessException{
48    List<ImageFileVO> imageList = (ArrayList)sqlSession.selectList("mapper.goods.selectGoodsDetailImage",goods_id);
49    return imageList;
50}
51@ Override//상품 사이즈 목록
52public List<String> selectGoodsSize(String goods_id) throws DataAccessException {
53    ArrayList list = (ArrayList)sqlSession.selectList("mapper.goods.selectGoodsSize",goods_id);
54    return list;
55}
56@ Override// 상품 종류별로 보는 페이지에서 가져오는 목록
57public List<GoodsVO> selectBySort(String goods_sort) throws DataAccessException {
58    List<GoodsVO> goodsList=(ArrayList)sqlSession.selectList("mapper.goods.selectBySort",goods_sort);
59    return goodsList;
60}
61
62}
```

DB(DataBase)를 처리하는 persistence영역 로직

# mapper\_goods(1)

```

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<!--<mapper namespace="mapper.goods">
    <resultMap type="goodsVO" id="goodsResult">
        <id property="goods_id" column="goods_id" />
        <result property="goods_sort" column="goods_sort" />
        <result property="goods_title" column="goods_title" />
        <result property="goods_price" column="goods_price" />
        <result property="goods_manufacturer" column="goods_manufacturer"/>
        <result property="goods_sales_price" column="goods_sales_price" />
        <result property="goods_point" column="goods_point" />
        <result property="goods_production_date" column="goods_production_date" />
        <result property="goods_delivery_price" column="goods_delivery_price" />
        <result property="goods_delivery_date" column="goods_delivery_date" />
        <result property="goods_status" column="goods_status" />
        <result property="goods_intro" column="goods_intro" />
        <result property="goods_enter_date" column="goods_enter_date" />
        <result property="goods_size" column="goods_size" />
        <result property="goods_color" column="goods_color" />
        <result property="goods_fileName" column="fileName" />
        <result property="goods_qty" column="goods_qty" />
    </resultMap>

    <resultMap type="imageFileVO" id="imageResult">
        <id property="image_id" column="image_id" />
        <result property="goods_id" column="goods_id" />
        <result property="fileName" column="filename" />
        <result property="reg_id" column="reg_id" />
        <result property="fileType" column="filetype" />
        <!-- <result property="reg_date" column="reg_date" /> -->
    </resultMap>
-->
```

- <mapper> 엘리먼트의 namespace는 유일한 SQL 아이디를 만들 때 각각의 SQL문의 id 와 조합하여 사용 (DAO클래스)
- resultMap : 검색 결과를 특정 자바 객체에 매칭 하여 리턴 할 때 join이나 검색 결과의 컬럼 값이 다를 때 매팅 하기 위한 방법

# mapper\_goods(2)

```

<!-- goodsStatus의 값으로 '신상품', '베스트셀러', '스테디셀러'를 전달하여 15개까지 검색 -->
<!-- 상품 리스트 보기 (15개 검색) - 조인(goods, goods_image) -->
<select id="selectGoodsList" parameterType="String" resultMap="goodsResult">
    <![CDATA[
        select t.* from (
            select g.*, i.filename
            from goods g, goods_image i
            where g.goods_id = i.goods_id
            and i.filetype = 'main_image'
            and g.goods_status = #{goodsStatus}
            order by g.goods_enter_date desc) t
        where rowNum < 16
    ]]>
</select>

<!-- 상품 상세 보기 (1건) - 조인(goods, goods_image) -->
<select id="selectGoodsDetail" parameterType="String" resultMap="goodsResult">
    <![CDATA[
        select g.*, i.filename
        from goods g, goods_image i
        where g.goods_id = i.goods_id
        and i.filetype = 'main_image'
        and g.goods_id = #{goods_id}
        order by g.goods_id
    ]]>
</select>
<!-- 상품별 사이즈 목록 -->
<select id="selectGoodsSize" parameterType="String" resultMap="goodsResult">
    <![CDATA[
        SELECT D.GOODS_SIZE
        FROM GOODS G, GOODS_DETAIL D
        WHERE G.GOODS_ID = D.GOODS_ID
        AND G.GOODS_ID = #{goods_id}
        ORDER BY G.GOODS_ID
    ]]>
</select>

```

- <select> 엘리먼트 : select 구문 작성하는 요소
- resultType : SQL결과를 resultMap과 매팅 하기 위한 속성

# mapper\_goods(2)

```

<!-- 상품 이미지 보기 (1건) -->
<select id="selectGoodsDetailImage" parameterType="String" resultMap="imageResult">
    <![CDATA[
        select * from goods_image
        where filetype != 'main_image'
        and goods_id = #{goods_id}
    ]]>
</select>

<!-- 상품제목 키워드로 상품 검색 - 조인(goods, goods_image) -->
<select id="selectGoodsBySearchWord" parameterType="String" resultMap="goodsResult">
    <![CDATA[
        select g.*, i.filename
        from goods g, goods_image i
        where g.goods_id = i.goods_id
        and i.filetype = 'main_image'
        and g.goods_title like '%'|| #{searchWord} || '%'
        order by g.goods_enter_date desc
    ]]>
</select>

<!-- 상품제목 키워드로 상품제목만 검색 -->
<select id="selectKeywordSearch" parameterType="String" resultType="String">
    <![CDATA[
        select goods_title from goods
        where goods_title like '%'|| #{keyword} || '%'
        order by goods_enter_date desc
    ]]>
</select>

<!-- 상품 종류별 목록 -->
<select id="selectBySort" parameterType="String" resultMap="goodsResult">
    <![CDATA[
        select t.* from (
            select g.*, i.filename
            from goods g, goods_image i
            where g.goods_id = i.goods_id
            and i.filetype = 'main_image'
            and g.goods_sort = #{goods_sort}
            order by g.goods_enter_date desc) t
        where rowNum < 16
    ]]>

```

```
@Component("memberVO")
public class MemberVO {
    private String member_id;
    private String member_pw;
    private String member_name;
    private String member_gender;
    private String tel1;
    private String tel2;
    private String tel3;
    private String hp1;
    private String hp2;
    private String hp3;
    private String sms_yn;
    private String email1;
    private String email2;
    private String email_yn;
    private String zipcode;
    private String road_address;
    private String jibun_address;
    private String detail_address;
    private String birth_year;
    private String birth_month;
    private String birth_day;
    private Date join_date;
    private String delete_yn;
```

- 회원 정보 데이터를 처리 시 사용하는 VO(Value Object)

# MemberController(1)

```
3 import java.util.Map;
22 @Controller("memberController")
23 @RequestMapping(value="/member")
24 public class MemberControllerImpl extends BaseController implements MemberController{
25     @Autowired
26     MemberService memberService;
27     @Autowired
28     MemberVO memberVO;
29
30     @Override
31     @RequestMapping(value="/login.do", method=RequestMethod.POST)
32     public ModelAndView login(@RequestParam Map<String, String> loginMap, HttpServletRequest request, HttpServletResponse response)
33         throws Exception {
34         ModelAndView mav = new ModelAndView();
35         memberVO = memberService.login(loginMap);
36         if(memberVO != null && memberVO.getMember_id() != null) {
37             HttpSession session = request.getSession();
38             session = request.getSession();
39             session.setAttribute("isLogOn", true);
40             session.setAttribute("memberInfo", memberVO);
41             System.out.println(memberVO);
42             String action = (String)session.getAttribute("action");
43             if(action != null && action.equals("/order/orderEachGoods.do")) {
44                 mav.setViewName("forward:" + action);
45             } else {
46                 mav.setViewName("redirect:/main/main.do");
47             }
48
49         }else {
50             String message = "아이디나 비밀번호가 다릅니다. 다시 로그인 해주세요.";
51             mav.addObject("message", message);
52             mav.setViewName("/member/loginForm");
53         }
54     return mav;
55 }
```

- 로그인 버튼 클릭 시 요청되는 controller

# MemberController(2)

- 로그아웃 클릭 시 요청되는 controller

```
@RequestMapping(value="/logout.do", method=RequestMethod.GET)
public ModelAndView logout(HttpServletRequest request, HttpServletResponse response) throws Exception {
    ModelAndView mav = new ModelAndView();
    HttpSession session = request.getSession();
    session.setAttribute("isLogOn", false);
    session.removeAttribute("memberInfo");
    session.removeAttribute("quickGoodsList");
    mav.setViewName("redirect:/main/main.do");
    return mav;
}
```

- 회원 정보와, quickMenu 리스트 제거

- 회원가입 클릭 시 요청되는 controller

```
@RequestMapping(value="/insertMember.do", method=RequestMethod.POST)
public ResponseEntity insertMember(MemberVO memberVO, HttpServletRequest request, HttpServletResponse response
    throws Exception {
    response.setContentType("text/html; charset=UTF-8");
    request.setCharacterEncoding("utf-8");
    String message = null;
    ResponseEntity resEntity = null;
    HttpHeaders responseHeaders = new HttpHeaders();
    responseHeaders.add("Content-Type", "text/html; charset=utf-8");
    try {
        memberService.insertMember(memberVO);
        message = "<script>";
        message += "alert('회원 가입을 마쳤습니다. 로그인창으로 이동합니다.')";
        message += "location.href='"+request.getContextPath() + "/member/memberForm.do'";
        message += "</script>";
    } catch (Exception e) {
        message = "<script>";
        message += "alert('작업 중 오류가 발생 했습니다. 다시 시도해 주세요.'";
        message += "location.href='"+request.getContextPath() + "/member/memberForm.do'";
        message += "</script>";
        e.printStackTrace();
    }
    resEntity = new ResponseEntity(message, responseHeaders, HttpStatus.OK);
    return resEntity;
}
```

## ResponceEntity

- Client(화면단)에 상태코드, HttpHeaders, 응답메시지, 반환 데이터를 모두 지정해서 반환 해주기 위해 사용하는 것

# MemberController(3)

```
@Override  
@RequestMapping(value="/overlapped.do", method= RequestMethod.POST)  
public ResponseEntity overlapped(@RequestParam("id")String id, HttpServletRequest request, HttpServletResponse response)  
throws Exception {  
    ResponseEntity resEntity = null;  
    String result = memberService.overlapped(id);  
    resEntity = new ResponseEntity(result, HttpStatus.OK);  
    return resEntity;  
}
```

- 회원가입 시 중복 계정 확인 클릭 시 요청되는 controller
- 중복 계정 있을 시 result값 보내기 위해 ResponseEntity 사용

# MemberServiceImpl

```
@Service("memberServices")
@Transactional(propagation=Propagation.REQUIRED)
public class MemberServiceImpl implements MemberService{
    @Autowired
    MemberDAO memberDAO;

    @Override
    public MemberVO login(Map loginMap) throws Exception {
        return memberDAO.login(loginMap);
    }

    @Override
    public void insertMember(MemberVO memberVO) throws Exception {
        memberDAO.insertMember(memberVO);
    }

    @Override
    public String overlapped(String id) throws Exception {
        return memberDAO.selectOverlappedID(id);
    }
}
```

MemberController에서 호출하는 비즈니스 로직

# MemberDAOImpl

```
@Service("memberServices")
@Transactional(propagation=Propagation.REQUIRED)
public class MemberServiceImpl implements MemberService{
    @Autowired
    MemberDAO memberDAO;

    @Override
    public MemberVO login(Map loginMap) throws Exception {
        return memberDAO.login(loginMap);
    }

    @Override
    public void insertMember(MemberVO memberVO) throws Exception {
        memberDAO.insertMember(memberVO);
    }

    @Override
    public String overlapped(String id) throws Exception {
        return memberDAO.selectOverlappedID(id);
    }
}
```

- DB(DataBase)를 처리하는 persistence 영역 로직

# mapper\_member(1)

```

6<!-- mapper namespace="mapper.member">
7<!-- 리절트 맵 정의 -->
8<resultMap id="memberResult" type="MemberVO">
9    <result property="member_id" column="member_id" />
10   <result property="member_pw" column="member_pw" />
11   <result property="member_name" column="member_name" />
12   <result property="member_gender" column="member_gender" />
13
14   <result property="birth_year" column="birth_year" />
15   <result property="birth_month" column="birth_month" />
16   <result property="birth_day" column="birth_day" />
17
18   <result property="tel1" column="tel1" />
19   <result property="tel2" column="tel2" />
20   <result property="tel3" column="tel3" />
21
22   <result property="hp1" column="hp1" />
23   <result property="hp2" column="hp2" />
24   <result property="hp3" column="hp3" />
25   <result property="sms_yn" column="sms_yn" />
26   <result property="email1" column="email1" />
27   <result property="email2" column="email2" />
28   <result property="email_yn" column="email_yn" />
29
30   <result property="zipcode" column="zipcode" />
31   <result property="road_address" column="road_address" />
32   <result property="jibun_address" column="jibun_address" />
33   <result property="detail_address" column="detail_address" />
34   <result property="join_date" column="join_date" />
35</resultMap>
36
37<!-- 로그인 시 계정과 비밀번호로 조회 sql-->
38<select id="login" resultType="memberVO" parameterType="java.util.Map" >
39  <![CDATA[
40    select *
41      from member
42      where member_id=#{member_id}
43      and member_pw=#{member_pw}
44
45  ]]>
46</select>

```

- 회원가입, 로그인, 로그아웃 처리하는 sql mapper

# mapper\_member(2)

```
48      <!-- 회원 가입시 중복계정 검색 sql-->
49@     <select id="selectOverlappedID" parameterType="String" resultType="String">
50         select decode(count(*),1, 'true', 0, 'false')
51         from member
52         where member_id = #{id}
53     </select>
54     <!-- 회원가입 sql -->
55@     <insert id="insertMember" parameterType="memberVO">
56     <!CDATA[
57         insert into member(member_id
58                         ,member_pw
59                         ,member_name
60                         ,member_gender
61                         ,birth_year
62                         ,birth_month
63                         ,birth_day
64                         ,tel1
65                         ,tel2
66                         ,tel3
67                         ,hp1
68                         ,hp2
69                         ,hp3
70                         ,sms_yn
71                         ,email1
72                         ,email2
73                         ,email_yn
74                         ,zipcode
75                         ,road_address
76                         ,jibun_address
77                         ,detail_address)
78         values(#{member_id}
79                         ,#{member_pw}
80                         ,#{member_name}
81                         ,#{member_gender}
82                         ,#{birth_year}
83                         ,#{birth_month}
84                         ,#{birth_day}
85                         ,#{tel1}
86                         ,#{tel2}
87                         ,#{tel3}
88                         ,#{hp1}
```

- 회원가입, 로그인, 로그아웃 처리하는 sql mapper

# mapper\_member(2)

```
48      <!-- 회원 가입시 중복계정 검색 sql-->
49@     <select id="selectOverlappedID" parameterType="String" resultType="String">
50         select decode(count(*),1, 'true', 0, 'false')
51         from member
52         where member_id = #{id}
53     </select>
54     <!-- 회원가입 sql -->
55@     <insert id="insertMember" parameterType="memberVO">
56     <!CDATA[
57         insert into member(member_id
58                         ,member_pw
59                         ,member_name
60                         ,member_gender
61                         ,birth_year
62                         ,birth_month
63                         ,birth_day
64                         ,tel1
65                         ,tel2
66                         ,tel3
67                         ,hp1
68                         ,hp2
69                         ,hp3
70                         ,sms_yn
71                         ,email1
72                         ,email2
73                         ,email_yn
74                         ,zipcode
75                         ,road_address
76                         ,jibun_address
77                         ,detail_address)
78         values(#{member_id}
79                         ,#{member_pw}
80                         ,#{member_name}
81                         ,#{member_gender}
82                         ,#{birth_year}
83                         ,#{birth_month}
84                         ,#{birth_day}
85                         ,#{tel1}
86                         ,#{tel2}
87                         ,#{tel3}
88                         ,#{hp1}
```

- 회원가입, 로그인, 로그아웃 처리하는 sql mapper

```
import java.sql.Date;[]

@Component("cartVO")
public class CartVO {
    private int cart_id;
    private int goods_id;
    private String member_id;
    private int cart_goods_qty;
    private Date reg_date;
    private String cart_goods_size;
```

회원 정보 데이터를 처리 시 사용하는 VO(Value Object)

# CartControllerImpl(1)

```

23 @Controller("cartController")
24 @RequestMapping(value="/cart")
25 public class CartControllerImpl extends BaseController implements CartController{
26     @Autowired
27     CartService cartService;
28     @Autowired
29     CartVO cartVO;
30     @Autowired
31     MemberVO memberVO;
32
33     @RequestMapping(value="/myCartList.do", method=RequestMethod.GET)
34     public ModelAndView myCartList(HttpServletRequest request, HttpServletResponse response) throws Exception {
35         String viewName = (String)request.getAttribute("viewName");
36         ModelAndView mav = new ModelAndView(viewName);
37         HttpSession session = request.getSession();
38         MemberVO memberVO = (MemberVO)session.getAttribute("memberInfo");
39         String member_id = memberVO.getMember_id();
40         cartVO.setMember_id(member_id);
41         Map<String, List> cartMap = cartService.myCartList(cartVO);
42         session.setAttribute("cartMap", cartMap);
43         return mav;
44     }
45
46     @RequestMapping(value="/addGoodsInCart.do", method=RequestMethod.POST, produces="application/text; charset=utf8")
47     public @ResponseBody String addGoodsInCart(@RequestParam("goods_id")int goods_id, @RequestParam("order_goods_qty")int order_goods_qty,
48                                                 @RequestParam("order_goods_size")String order_goods_size,
49                                                 HttpServletRequest request, HttpServletResponse response)
50         throws Exception {
51         HttpSession session = request.getSession();
52         memberVO = (MemberVO)session.getAttribute("memberInfo");
53         String member_id = memberVO.getMember_id();
54
55         cartVO.setMember_id(member_id);
56         cartVO.setGoods_id(goods_id);
57         cartVO.setCart_goods_qty(order_goods_qty);
58         cartVO.setCart_goods_size(order_goods_size);
59         boolean isAlreadyExisted = cartService.findCartGoods(cartVO);
60         if(isAlreadyExisted == true) {
61             return "already_existed";
62         } else {
63             cartService.addGoodsInCart(cartVO);
64             return "add_success";
65         }
66     }

```

- 장바구니 리스트 요청 시 작동 하는 controller
- 제품 상세 화면에서 장바구니 담기 클릭 시 동작 하는 controller

- @RequestParam("") : jsp에서 값을 받아 올 수 있는 어노테이션
- 장바구니 테이블에 넣을 데이터를 넘겨 받음

# CartControllerImpl(2)

- 장바구니 리스트에서 수량 변경 요청 시 동작하는 controller

```
@RequestMapping(value="/modifyCartQty.do", method = RequestMethod.POST)
public @ResponseBody String modifyCartQty(@RequestParam("goods_id")int goods_id,
                                         @RequestParam("cart_goods_qty")int cart_goods_qty,
                                         HttpServletRequest request, HttpServletResponse response) throws Exception {
    HttpSession session = request.getSession();
    memberVO = (MemberVO)session.getAttribute("memberInfo");
    String member_id = memberVO.getMember_id();
    cartVO.setGoods_id(goods_id);
    cartVO.setMember_id(member_id);
    cartVO.setCart_goods_qty(cart_goods_qty);
    boolean result = cartService.modifyCartQty(cartVO);

    if(result == true) {
        return "modify_success";
    } else {
        return "modify_failed";
    }
}
```

- 장바구니 리스트에서 삭제 요청 시 동작하는 controller

```
@RequestMapping(value="/removeCartGoods.do", method=RequestMethod.POST)
public ModelAndView removeCartGoods(@RequestParam("cart_id")int cart_id, HttpServletRequest request, HttpServletResponse response)
    throws Exception {
    ModelAndView mav = new ModelAndView();
    cartService.removeCartGoods(cart_id);
    mav.setViewName("redirect:/cart/myCartList.do");
    return mav;
}
```

```

16 @Service("cartService")
17 @Transactional(propagation=Propagation.REQUIRED)
18 public class CartServiceImpl implements CartService{
19     @Autowired
20     CartDAO cartDAO;
21
22     @Override
23     public Map<String, List> myCartList(CartVO cartVO) throws Exception {
24         Map<String, List> cartMap = new HashMap<String, List>();
25         List<CartVO> myCartList = cartDAO.selectCartList(cartVO);
26         if(myCartList.size()==0) {
27             return null;
28         }
29         List<GoodsVO> myGoodsList = cartDAO.selectGoodsList(myCartList);
30         cartMap.put("myCartList", myCartList);
31         cartMap.put("myGoodsList", myGoodsList);
32         return cartMap;
33     }
34
35     @Override
36     public boolean findCartGoods(CartVO cartVO) throws Exception {
37         return cartDAO.selectCountInCart(cartVO);
38     }
39
40     @Override
41     public void addGoodsInCart(CartVO cartVO) throws Exception {
42         cartDAO.insertGoodsInCart(cartVO);
43     }
44
45     @Override
46     public boolean modifyCartQty(CartVO cartVO) throws Exception {
47         boolean result = true;
48         cartDAO.updateCartGoodsQty(cartVO);
49         return result;
50     }
51
52     @Override
53     public void removeCartGoods(int cart_id) throws Exception {
54         cartDAO.deleteCartGoods(cart_id);
55     }
56
57 }

```

- CartController에서 호출하는 비즈니스 로직

- 장바구니 리스트 화면에서 보여줄 수량과, 상품 목록을 보여주기 위해 각각 따로 map에 저장

# cartDAOImpl

```

@Repository("cartDAO")
public class CartDAOImpl implements CartDAO{
    @Autowired
    private SqlSession sqlSession;

    @Override
    public List<CartVO> selectCartList(CartVO cartVO) throws DataAccessException {
        List<CartVO> cartList = (List)sqlSession.selectList("mapper.cart.selectCartList", cartVO);
        return cartList;
    }

    @Override
    public List<GoodsVO> selectGoodsList(List<CartVO> cartList) throws DataAccessException {
        List<GoodsVO> myGoodsList;
        System.out.println(cartList.toString());
        myGoodsList = sqlSession.selectList("mapper.cart.selectGoodsList", cartList);
        return myGoodsList;
    }

    @Override
    public boolean selectCountInCart(CartVO cartVO) throws DataAccessException {
        String result = sqlSession.selectOne("mapper.cart.selectCountInCart", cartVO);
        return Boolean.parseBoolean(result);
    }

    @Override
    public void insertGoodsInCart(CartVO cartVO) throws DataAccessException {
        int cart_id = selectMaxCartId();
        cartVO.setCart_id(cart_id);
        sqlSession.insert("mapper.cart.insertGoodsInCart", cartVO);
    }

    @Override
    public void updateCartGoodsQty(CartVO cartVO) throws DataAccessException {
        sqlSession.insert("mapper.cart.updateCartGoodsQty", cartVO);
    }

    @Override
    public void deleteCartGoods(int cart_id) throws DataAccessException {
        sqlSession.delete("mapper.cart.deleteCartGoods", cart_id);
    }
}

```

- 장바구니 id 생성

```

private int selectMaxCartId() throws DataAccessException{
    int cart_id = sqlSession.selectOne("mapper.cart.selectMaxCartId");
    return cart_id;
}

```

DB(DataBase)를 처리하는 persistence 영역 로직

# mapper\_cart(1)

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dt
3
4<mapper namespace="mapper.cart">
5
6    <resultMap type="cartVO" id="cartResult">
7        <id property="cart_id" column="cart_id" />
8        <result property="goods_id" column="goods_id" />
9        <result property="member_id" column="member_id" />
10       <result property="reg_date" column="reg_date" />
11       <result property="cart_goods_qty" column="cart_goods_qty" />
12   </resultMap>
13
14    <resultMap type="goodsVO" id="goodsResult">
15        <id property="goods_id" column="goods_id" />
16        <result property="goods_sort" column="goods_sort" />
17        <result property="goods_title" column="goods_title" />
18        <result property="goods_manufacturer" column="goods_manufacturer" />
19        <result property="goods_production_date" column="goods_production_date" />
20
21        <result property="goods_price" column="goods_price" />
22        <result property="goods_sales_price" column="goods_sales_price" />
23        <result property="goods_delivery_price" column="goods_delivery_price" />
24        <result property="goods_delivery_date" column="goods_delivery_date" />
25
26        <result property="goods_status" column="goods_status" />
27        <result property="goods_intro" column="goods_intro" />
28        <result property="goods_size" column="goods_size" />
29        <result property="goods_color" column="goods_color" />
30        <result property="goods_qty" column="goods_qty" />
31        <result property="goods_fileName" column="filename" />
32    </resultMap>
33
34    <!-- 아이디로 장바구니 정보 확인 -->
35    <select id="selectCartList" resultMap="cartResult">
36        <![CDATA[
37            SELECT *
38            FROM CART
39            WHERE MEMBER_ID=#{member_id}
40            order by cart_id
41        ]]>
42    </select>
```

- 장바구니 화면에서 사용하는 sql mapper

# mapper\_cart(2)

```

<!-- 상품리스트를 입고일자 내림차순으로 조회 -->
<select id="selectGoodsList" parameterType="java.util.Map" resultMap="goodsResult">
    <![CDATA[
        SELECT G.*
            , I.FILENAME
        FROM GOODS G
            , GOODS_IMAGE I
            , cart c
        WHERE G.GOODS_ID = I.GOODS_ID
        and i.goods_id = c.goods_id
            AND I.FILETYPE = 'main_image'
            AND G.GOODS_ID IN
    ]]>
    <foreach item="item" collection="list" open="(" separator="," close=")">
        #{item.goods_id}
    </foreach>
    order by cart_id
</select>
<!-- 상품아이디와 회원아이디로 장바구니 유무 조회 -->
<select id="selectCountInCart" parameterType="cartVO" resultType="String">
    <![CDATA[
        SELECT DECODE(COUNT(*), 0, 'false','true')
        FROM CART
        WHERE GOODS_ID=#{goods_id}
            AND MEMBER_ID=#{member_id}
    ]]>
</select>
<!-- 장바구니 테이블에 상품 정보 저장 -->
<insert id="insertGoodsInCart" parameterType="cartVO">
    <![CDATA[
        INSERT INTO CART(CART_ID
            , GOODS_ID
            , MEMBER_ID
            , CART_GOODS_SIZE
            , CART_GOODS_QTY)
        VALUES/#{cart_id}
            , #{goods_id}
            , #{member_id}
            , #{cart_goods_size}
            , #{cart_goods_qty})
    ]]>
</insert>

```

- 장바구니 화면에서 사용하는 sql mapper

## mapper\_cart(3)

```
<!-- 장바구니 수량 수정 -->
<update id="updateCartGoodsQty" parameterType="cartVO">
    <![CDATA[
        UPDATE CART
        SET CART_GOODS_QTY=#{cart_goods_qty}
        WHERE MEMBER_ID=#{member_id}
        AND GOODS_ID=#{goods_id}
    ]]>
</update>

<delete id="deleteCartGoods" parameterType="int">
    <![CDATA[
        DELETE
        FROM CART
        WHERE CART_ID=#{cart_id}
    ]]>
</delete>
<!-- 장바구니 아이디 시퀀스 생성-->
<select id="selectMaxCartId" resultType="int">
    <![CDATA[
        SELECT COUNT(NVL(CART_ID, 0))+1
        FROM CART
    ]]>
</select>
```

- 장바구니 화면에서 사용하는 sql mapper

# mypageControllerImpl(1)

```

@RequestMapping(value="/mypage")
public class MyPageControllerImpl extends BaseController implements MyPageController{
    @Autowired
    MyPageService myPageService;
    @Autowired
    MemberVO memberVO;

    마이페이지 요청 시 작동 하는 controller

    @RequestMapping(value="/myPageMain.do" ,method = RequestMethod.GET)
    public ModelAndView myPageMain(@RequestParam(required = false,value="message") String message,
                                   HttpServletRequest request, HttpServletResponse response) throws Exception {
        HttpSession session=request.getSession();
        session=request.getSession();
        session.setAttribute("side_menu", "my_page");

        String viewName=(String)request.getAttribute("viewName");
        ModelAndView mav = new ModelAndView(viewName);
        memberVO=(MemberVO)session.getAttribute("memberInfo");
        String member_id=memberVO.getMember_id();

        List<OrderVO> myOrderList=myPageService.listMyOrderGoods(member_id);

        mav.addObject("message", message);
        mav.addObject("myOrderList", myOrderList);

        return mav;
    }

    주문 번호 클릭(상세 보기) 시 동작 하는 controller

    @RequestMapping(value="/myOrderDetail.do" ,method = RequestMethod.GET)
    public ModelAndView myOrderDetail(@RequestParam("order_id") String order_id,
                                      HttpServletRequest request,
                                      HttpServletResponse response) throws Exception {
        String viewName=(String)request.getAttribute("viewName");
        ModelAndView mav = new ModelAndView(viewName);
        HttpSession session=request.getSession();
        MemberVO orderer=(MemberVO)session.getAttribute("memberInfo");

        List<OrderVO> myOrderList=myPageService.findMyOrderInfo(order_id);
        mav.addObject("orderer", orderer);
        mav.addObject("myOrderList",myOrderList);
        return mav;
    }
}

```

사이드 메뉴 리스트를 변화  
하기 위한 플래그 저장

```

@RequestMapping(value="/cancelMyOrder.do" ,method = RequestMethod.POST)
public ModelAndView cancelMyOrder(@RequestParam("order_id") String order_id,
                                   HttpServletRequest request,
                                   HttpServletResponse response) throws Exception {
    ModelAndView mav = new ModelAndView();
    myPageService.cancelOrder(order_id);
    mav.addObject("message", "cancel_order");
    mav.setViewName("redirect:/mypage/myPageMain.do");
    return mav;
}

```

- 주문 취소 버튼 클릭 시 동작 하는 controller
- 자바스크립트에 message 전달

# mypageControllerImpl(2)

```

@RequestMapping(value="/listMyOrderHistory.do" ,method = RequestMethod.GET)
public ModelAndView listMyOrderHistory(@RequestParam Map<String, String> dateMap,
                                         HttpServletRequest request,
                                         HttpServletResponse response) throws Exception {
    String viewName=(String)request.getAttribute("viewName");
    ModelAndView mav = new ModelAndView(viewName);
    HttpSession session=request.getSession();
    memberVO=(MemberVO)session.getAttribute("memberInfo");
    String member_id=memberVO.getMember_id();

    String fixedSearchPeriod = dateMap.get("fixedSearchPeriod");
    String search_type = dateMap.get("search_type");
    String search_word = dateMap.get("search_word");
    String beginDate=dateMap.get("beginDate");
    String endDate=dateMap.get("endDate");

    if(beginDate == null) {
        String [] tempDate=calcSearchPeriod(fixedSearchPeriod).split(",");
        beginDate=tempDate[0];
        endDate=tempDate[1];
    }
    dateMap.put("beginDate", beginDate);
    dateMap.put("endDate", endDate);
    dateMap.put("member_id", member_id);
    dateMap.put("search_type",search_type);
    dateMap.put("search_word", search_word);
    List<OrderVO> myOrderHistList=myPageService.listMyOrderHistory(dateMap);

    String beginDate1[]=beginDate.split("-");
    String endDate1[]=endDate.split("-");
    mav.addObject("beginYear",beginDate1[0]);
    mav.addObject("beginMonth",beginDate1[1]);
    mav.addObject("beginDay",beginDate1[2]);
    mav.addObject("endYear",endDate1[0]);
    mav.addObject("endMonth",endDate1[1]);
    mav.addObject("endDay",endDate1[2]);
    mav.addObject("myOrderHistList", myOrderHistList);
    return mav;
}

```

주문내역/배송 조회 요청 시 동작 하는 controller

고정 기간 조회 시 받아오는 날짜

상세 조회 시 받아오는 날짜

검색 조건 map에 저장

"-" 기준으로 날짜 잘라서 전송

# mypageControllerImpl(3)

```

@Override
@RequestMapping(value="/myDetailInfo.do" ,method = RequestMethod.GET)
public ModelAndView myDetailInfo(HttpServletRequest request,
    HttpServletResponse response) throws Exception {
    String viewName=(String)request.getAttribute("viewName");
    ModelAndView mav = new ModelAndView(viewName);
    return mav;
}

```

수정 버튼 클릭시 요청 되는 controller

```

@RequestMapping(value="/modifyMyInfo.do" ,method = RequestMethod.POST)
public ResponseEntity modifyMyInfo(@RequestParam("attribute") String attribute,
    @RequestParam("value") String value,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception {
    Map<String, String> memberMap=new HashMap<String, String>();
    String val[] = null;
    HttpSession session=request.getSession();
    memberVO=(MemberVO)session.getAttribute("memberInfo");
    String member_id=memberVO.getMember_id();
    if(attribute.equals("member_birth")){
        val=value.split(",");
        memberMap.put("birth_year",val[0]);
        memberMap.put("birth_month",val[1]);
        memberMap.put("birth_day",val[2]);
    }else if(attribute.equals("tel")){
        val=value.split(",");
        memberMap.put("tel1",val[0]);
        memberMap.put("tel2",val[1]);
        memberMap.put("tel3",val[2]);
    }else if(attribute.equals("hp")){
        val=value.split(",");
        memberMap.put("hp1",val[0]);
        memberMap.put("hp2",val[1]);
        memberMap.put("hp3",val[2]);
        memberMap.put("sms_yn", val[3]);
    }else if(attribute.equals("email")){
        val=value.split(",");
        memberMap.put("email1",val[0]);
        memberMap.put("email2",val[1]);
        memberMap.put("email_yn", val[2]);
    }
}

```

- 회원정보 관리 클릭 시 동작 하는 controller
- session에 memberVO 데이터가 있기 때문에 화면으로 이동

```

@RequestMapping(value="/cancelMyOrder.do" ,method = RequestMethod.POST)
public ModelAndView cancelMyOrder(@RequestParam("order_id") String order_id,
    HttpServletRequest request,
    HttpServletResponse response) throws Exception {
    ModelAndView mav = new ModelAndView();
    myPageService.cancelOrder(order_id);
    mav.addObject("message", "cancel_order");
    mav.setViewName("redirect:/mypage/myPageMain.do");
    return mav;
}

```

각 항목 별로 수정 후 다시 전달

```

} else {
    memberMap.put(attribute,value);
}
memberMap.put("member_id", member_id);
memberVO=(MemberVO)myPageService.modifyMyInfo(memberMap);
session.removeAttribute("memberInfo");
session.setAttribute("memberInfo", memberVO);

String message = null;
ResponseEntity resEntity = null;
HttpHeaders responseHeaders = new HttpHeaders();
message = "mod_success";
resEntity =new ResponseEntity(message, responseHeaders, HttpStatus.OK);
return resEntity;
}

```

# mypageServiceImpl

```
@Service("myPageService")
@Transactional(propagation=Propagation.REQUIRED)
public class MyPageServiceImpl implements MyPageService{
    @Autowired
    MyPageDAO myPageDAO;

    public List<OrderVO> listMyOrderGoods(String member_id) throws Exception{
        return myPageDAO.selectMyOrderGoodsList(member_id);
    }

    public List<OrderVO> findMyOrderInfo(String order_id) throws Exception{
        return myPageDAO.selectMyOrderInfo(order_id);
    }

    public List<OrderVO> listMyOrderHistory(Map dateMap) throws Exception{
        return myPageDAO.selectMyOrderHistoryList(dateMap);
    }

    public MemberVO modifyMyInfo(Map memberMap) throws Exception{
        String member_id=(String)memberMap.get("member_id");
        myPageDAO.updateMyInfo(memberMap);
        return myPageDAO.selectMyDetailInfo(member_id);
    }

    public void cancelOrder(String order_id) throws Exception{
        myPageDAO.updateMyOrderCancel(order_id);
    }
    public MemberVO myDetailInfo(String member_id) throws Exception{
        return myPageDAO.selectMyDetailInfo(member_id);
    }
}
```

mypageController에서 호출 하는 비지니스 영역

# mypageDAOImpl

```
@Repository("myPageDAO")
public class MyPageDAOImpl implements MyPageDAO{
    @Autowired
    private SqlSession sqlSession;

    public List<OrderVO> selectMyOrderGoodsList(String member_id) throws DataAccessException{
        List<OrderVO> orderGoodsList=(List)sqlSession.selectList("mapper.mypage.selectMyOrderGoodsList",member_id);
        return orderGoodsList;
    }

    public List selectMyOrderInfo(String order_id) throws DataAccessException{
        List myOrderList=(List)sqlSession.selectList("mapper.mypage.selectMyOrderInfo",order_id);
        return myOrderList;
    }

    public List<OrderVO> selectMyOrderHistoryList(Map dateMap) throws DataAccessException{
        List<OrderVO> myOrderHistList=(List)sqlSession.selectList("mapper.mypage.selectMyOrderHistoryList",dateMap);
        return myOrderHistList;
    }

    public void updateMyInfo(Map memberMap) throws DataAccessException{
        sqlSession.update("mapper.mypage.updateMyInfo",memberMap);
    }

    public MemberVO selectMyDetailInfo(String member_id) throws DataAccessException{
        MemberVO memberVO=(MemberVO)sqlSession.selectOne("mapper.mypage.selectMyDetailInfo",member_id);
        return memberVO;
    }

    public void updateMyOrderCancel(String order_id) throws DataAccessException{
        sqlSession.update("mapper.mypage.updateMyOrderCancel",order_id);
    }
}
```

DB(DataBase)를 처리하는 persistence 영역 로직

# mapper\_mypage(1)

```
< mapper namespace="mapper.mypage">
<! -- 리절트 맵 정의 -->
<resultMap id="orderGoodsResult" type="orderVO">
    <result property="order_id" column="order_id" />
    <result property="member_id" column="member_id" />
    <result property="goods_id" column="goods_id" />
    <result property="goods_title" column="goods_title" />
    <result property="goods_sales_price" column="goods_sales_price" />
    <result property="order_goods_qty" column="order_goods_qty" />

    <result property="orderer_name" column="orderer_name" />
    <result property="receiver_name" column="receiver_name" />
    <result property="receiver_hp1" column="receiver_hp1" />
    <result property="receiver_hp2" column="receiver_hp2" />
    <result property="receiver_hp3" column="receiver_hp3" />

    <result property="receiver_tel1" column="receiver_tel1" />
    <result property="receiver_tel2" column="receiver_tel2" />
    <result property="receiver_tel3" column="receiver_tel3" />
    <result property="delivery_address" column="delivery_address" />
    <result property="delivery_message" column="delivery_message" />

    <result property="delivery_method" column="delivery_method" />
    <result property="gift_wrapping" column="gift_wrapping" />
    <result property="pay_method" column="pay_method" />
    <result property="card_com_name" column="card_com_name" />
    <result property="card_pay_month" column="card_pay_month" />

    <result property="pay_orderer_hp" column="pay_orderer_hp" />
    <result property="pay_order_time" column="pay_order_time" />
    <result property="delivery_state" column="delivery_state" />
    <result property="final_total_price" column="final_total_price" />
    <result property="goods_qty" column="goods_qty" />

    <result property="goods_fileName" column="goods_fileName" />
</resultMap>
```

아이디 별 주문 정보 조회

```
<select id="selectMyOrderGoodsList" resultMap="orderGoodsResult" parameterType="String" >
<! [CDATA[
    select * from orders
    where member_id=#{member_id}
    order by pay_order_time desc
]]>
</select>
```

주문 아이디로 주문 정보 조회

```
<select id="selectMyOrderInfo" resultMap="orderGoodsResult" parameterType="String" >
<! [CDATA[
    select * from orders
    where order_id=#{order_id}
]]>
</select>
<select id="selectMyOrderHistoryList" resultMap="orderGoodsResult" parameterType="java.util.Map" >
    select * from orders
    where member_id=#{member_id}
    and to_char(pay_order_time,'yyyy-mm-dd') between #{beginDate} and #{endDate}
    <if test="search_type=='receiver_name'">
        AND receiver_name LIKE '%||#{search_word}||%'
    </if>
    <if test="search_type=='member_id'">
        AND member_id LIKE '%||#{search_word}||%'
    </if>
    <if test="search_type=='order_id'">
        AND order_id LIKE '%||#{search_word}||%'
    </if>
</select>
```

controller에서 map에 저장한 검색 조건에 매핑하여 검색 키워드 설정 후 조회

마이페이지에서 동작하는 sql

# mapper\_mypage(2)

```
<update id="updateMyInfo" parameterType="java.util.HashMap">
    update member
        <set>
            <if test=" member_pw!='' and member_pw!=null">
                member_pw=#{member_pw},
            </if>
            <if test=" member_gender!='' and member_gender!=null">
                member_gender=#{member_gender},
            </if>
            <if test=" birth_year!='' and birth_year!=null">
                birth_year=#{birth_year},
            </if>
            <if test=" birth_month!='' and birth_month!=null">
                birth_month=#{birth_month},
            </if>
            <if test=" birth_day!='' and birth_day!=null">
                birth_day=#{birth_day},
            </if>
            <if test=" tel1!='' and tel1!=null">
                tel1=#{tel1},
            </if>
            <if test=" tel2!='' and tel2!=null">
                tel2=#{tel2},
            </if>
            <if test=" tel3!='' and tel3!=null">
                tel3=#{tel3},
            </if>
            <if test=" hp1!='' and hp1!=null">
                hp1=#{hp1},
            </if>
            <if test=" hp2!='' and hp2!=null">
                hp2=#{hp2},
            </if>
            <if test=" hp3!='' and hp3!=null">
                hp3=#{hp3},
            </if>
        </set>
    where
        member_id=#{member_id}
</update>
```

map에 저장된 키워드에 따라 컬럼 별 각각의 데이터 수정

```
        sms_yn=#{sms_yn},
    </if>
    <if test=" email1!='' and email1!=null">
        email1=#{email1},
    </if>
    <if test=" email2!='' and email2!=null">
        email2=#{email2},
    </if>
    <if test=" email_yn!='' and email_yn!=null">
        email_yn=#{email_yn},
    </if>
    <if test=" zipcode!='' and zipcode!=null">
        zipcode=#{zipcode},
    </if>
    <if test=" road_address!='' and road_address!=null">
        road_address=#{road_address},
    </if>
    <if test=" jibun_address!='' and jibun_address!=null">
        jibun_address=#{jibun_address},
    </if>
    <if test=" detail_address!='' and detail_address!=null">
        detail_address=#{detail_address}
    </if>
</set>
where
    member_id=#{member_id}
</update>
```

마이페이지에서 동작하는 sql

```
<select id="selectMyDetailInfo" resultType="memberVO" parameterType="String" >
    <![CDATA[
        select * from member
        where member_id=#{member_id}
    ]]>
</select>

<update id="updateMyOrderCancel" parameterType="String">
    update orders
    set delivery_state='cancel_order'
    where order_id=#{order_id}
</update>
```

자신 정보 조회하는 sql

주문 취소 시 상태 변경

마이페이지에서 동작하는 sql

# AdminGoodsControllerImpl(1)

```

@Controller("adminGoodsController")
@RequestMapping(value="/admin/goods")
public class AdminGoodsControllerImpl extends BaseController implements AdminGoodsController{
    private static final String CURR_IMAGE_REPO_PATH = "/Users/zidol/Desktop/shopping/file_repo";
    @Autowired
    AdminGoodsService adminGoodsService;

    @RequestMapping(value="/adminGoodsMain.do", method= {RequestMethod.POST, RequestMethod.GET})
    public ModelAndView adminGoodsMain(Map<String, String> dateMap, HttpServletRequest request,
        HttpServletResponse response) throws Exception {
        String viewName = (String)request.getAttribute("viewName");
        ModelAndView mav = new ModelAndView(viewName);
        HttpSession session = request.getSession();
        session = request.getSession();
        session.setAttribute("side_menu", "admin_mode");

        String fixedSearchPeriod = dateMap.get("fixedSearchPeriod");
        String section = dateMap.get("section");
        String pageNum = dateMap.get("pageNum");
        String search_type = dateMap.get("search_type");
        String search_word = dateMap.get("search_word");
        String beginDate=dateMap.get("beginDate");
        String endDate=dateMap.get("endDate");

        if(beginDate == null) {
            String [] tempDate=calcSearchPeriod(fixedSearchPeriod).split(",");
            beginDate=tempDate[0];
            endDate=tempDate[1];
        }

        dateMap.put("beginDate", beginDate);
        dateMap.put("endDate", endDate);

        Map<String, Object> condMap = new HashMap<String, Object>();
        if(section == null) {
            section = "1";
        }
        condMap.put("section", section);
        if(pageNum ==null) {
            pageNum = "1";
        }
    }
}

```

관리자 메인 페이지 요청 시 작동 하는 controller

검색 키워드, 값 map에 저장 후 전달

```

condMap.put("pageNum", pageNum);
        put("beginDate", beginDate);
        put("endDate", endDate);
        put("search_type", search_type);
condMap.put("search_word", search_word);
List<GoodsVO> newGoodsList = adminGoodsService.listNewGoods(condMap);
mav.addObject("newGoodsList", newGoodsList);

String beginDate1[] = beginDate.split("-");
String endDate2[] = endDate.split("-");
mav.addObject("beginYear", beginDate1[0]);
mav.addObject("beginMonth", beginDate1[1]);
mav.addObject("beginDay", beginDate1[2]);
mav.addObject("endYear", endDate2[0]);
mav.addObject("endMonth", endDate2[1]);
mav.addObject("endDay", endDate2[2]);

mav.addObject("section", section);
mav.addObject("pageNum", pageNum);
return mav;
}

```

# AdminGoodsControllerImpl(2)

```

@RequestMapping(value="/addNewGoods.do", method = {RequestMethod.POST})
public ResponseEntity addNewGoods(MultipartHttpServletRequest multipartRequest, HttpServletResponse response)
    throws Exception {
    multipartRequest.setCharacterEncoding("utf-8");
    response.setContentType("text/html; charset=UTF-8");
    String imageFileName = null;

    Map newGoodsMap = new HashMap();
    List detail = new ArrayList();
    Enumeration enu = multipartRequest.getParameterNames();
    Enumeration enuDetail = multipartRequest.getParameterNames();
    String arr[] = null;

    while(enu.hasMoreElements()) {
        String name = (String) enu.nextElement();
        String value = multipartRequest.getParameter(name);
        newGoodsMap.put(name, value);
    }
    System.out.println(detail);
    HttpSession session = multipartRequest.getSession();
    MemberVO memberVO = (MemberVO)session.getAttribute("memberInfo");
    System.out.println(memberVO.getMember_id());
    String reg_id = memberVO.getMember_id();

    List<ImageFileVO>imageFileList = upload(multipartRequest);
    if(imageFileList != null && imageFileList.size() != 0) {
        for(ImageFileVO imageFileVO : imageFileList) {
            imageFileVO.setReg_id(reg_id);
        }
        newGoodsMap.put("imageFileList", imageFileList);
    }
}

```

관리자 새 상품 등록 페이지 요청 시 동작하는 controller

파일 업로드 때문에 MultipartServletRequest 사용

```

String message = null;
ResponseEntity resEntity = null;
HttpHeaders responseHeaders = new HttpHeaders();
responseHeaders.add("Content-Type", "text/html; charset=utf-8");
try {
    int goods_id = adminGoodsService.addNewGoods(newGoodsMap);
    Map<String, List> paramMap = new HashMap<String, List>();
    Map newGoodsDetailMap;
    while(enuDetail.hasMoreElements()) {
        String name = (String) enuDetail.nextElement();
        if(name.equals("goods_size")) {
            arr = multipartRequest.getParameterValues(name);
            for(int i=0; i < arr.length; i++) {
                newGoodsDetailMap = new HashMap();
                newGoodsDetailMap.put("goods_id", goods_id);
                newGoodsDetailMap.put("goods_color", multipartRequest.getParameter("goods_color"));
                newGoodsDetailMap.put("goods_atv", multipartRequest.getParameter("goods_atv"));
                newGoodsDetailMap.put(name, arr[i]);
                detail.add(newGoodsDetailMap);
            }
        }
    }
    paramMap.put("detailMap", detail);
    adminGoodsService.addNewGoodsDetail(paramMap);
    if(imageFileList != null && imageFileList.size() != 0) {
        for(ImageFileVO imageFileVO : imageFileList) {
            imageFileName = imageFileVO.getFileName();
            File srcFile = new File(CURR_IMAGE_REPO_PATH + "/" + "temp" + "/" + imageFileName);
            File destDir = new File(CURR_IMAGE_REPO_PATH + "/" + goods_id);
            FileUtils.moveFileToDirectory(srcFile, destDir, true);
        }
    }
    message = "<script>";
    message += "alert('성공적으로 입력 하였습니다');";
    message += "location.href='" + multipartRequest.getContextPath() + "/admin/goods/addNewGoodsForm.do';";
    message += ("</script>");
}

```

checkbox로 여러 사이즈를 입력하기 때문에 map에 키별로 저장 후 list에 저장

이미지 첨부 시 temp 폴더에 저장 후 상품 id 폴더 생성 후 이동

- enu : goods 테이블에 넣어야 할 데이터
- enuDetail : goods\_detail 테이블에 넣어야 할 데이터
- 각각 테이블에 저장할 데이터를 2개의 map에 저장

## AdminGoodsControllerImpl(2-1)

```
        } catch (Exception e) {
            if(imageFileList != null && imageFileList.size() != 0) {
                for(ImageFileVO imageFileVO : imageFileList) {
                    imageName = imageFileVO.getFileName();
                    File srcFile = new File(CURR_IMAGE_REPO_PATH + "\\\" + "temp" + "\\\" + imageName);
                    srcFile.delete();
                }
            }
            message = "<script>";
            message += "alert('등록 실패 하였습니다');";
            message += "location.href='" + multipartRequest.getContextPath() + "/admin/goods/addNewGoodsForm.do'";
            message += ("</script>");
            e.printStackTrace();
        }
        resEntity = new ResponseEntity(message, responseHeaders, HttpStatus.OK);
        return resEntity;
    }
}
```

예외 발생 시 업로드 한 이미지 파일 삭제  
자바스크립트에서 사용할 message, header 리턴

# AdminGoodsControllerImpl(3)

```

@RequestMapping(value="/modifyGoodsForm.do", method= {RequestMethod.GET, RequestMethod.POST})
public ModelAndView modifyGoodsForm(@RequestParam("goods_id")int goods_id, @RequestParam("goods_size")String goods_size,
                                     HttpServletRequest request,
                                     HttpServletResponse response) throws Exception {
    String viewName = (String) request.getAttribute("viewName");
    ModelAndView mav = new ModelAndView(viewName);
    Map condMap = new HashMap<>();
    condMap.put("goods_id", goods_id);
    condMap.put("goods_size", goods_size);
    Map goodsMap = adminGoodsService.goodsDetail(condMap);
    mav.addObject("goodsMap", goodsMap);
    return mav;
}

@RequestMapping(value="/modifyGoodsInfo.do", method= {RequestMethod.POST})
public ResponseEntity modifyGoodsInfo(@RequestParam("goods_id") String goods_id,
                                       @RequestParam(value="goods_size", required=false) String goods_size,
                                       @RequestParam(value="goods_color", required=false) String goods_color,
                                       @RequestParam("attribute")String attribute,
                                       @RequestParam("value")String value, HttpServletRequest request, HttpServletResponse response) throws Exception {
    Map<String, String> goodsMap = new HashMap<String, String>();
    goodsMap.put("goods_id", goods_id);
    goodsMap.put(attribute, value);
    System.out.println("goods_size: " + goods_size);
    System.out.println("goods_color: " + goods_color);
    System.out.println("goods_qty: " + attribute);
    if(attribute.equals("goods_qty")) {
        System.out.println("1");
        goodsMap.put("goods_size", goods_size);
        goodsMap.put("goods_color", goods_color);
        adminGoodsService.modifyGoodsDetailInfo(goodsMap);
    }else {
        System.out.println("2");
        adminGoodsService.modifyGoodsInfo(goodsMap);
    }

    String message = null;
    ResponseEntity resEntity = null;
    HttpHeaders responseHeaders = new HttpHeaders();
    message = "mod_success";
    resEntity = new ResponseEntity(message, responseHeaders, HttpStatus.OK);
    return resEntity;
}

```

- 상품 수정 페이지 호출 시 동작하는 controller
- goods\_detail 테이블 수정 위해 goods\_size를 받음
- 상품 정보를 보여주는 화면으로 넘어감

- 수정 버튼 클릭 시 동작 하는 controller
- attribute의 키값을 받아 goods, goods\_table의 컬럼을 정해서 수정

# AdminGoodsControllerImpl(4)

```

@RequestMapping(value="/modifyGoodsImageInfo.do" ,method={RequestMethod.POST})
public void modifyGoodsImageInfo(MultipartHttpServletRequest multipartRequest, HttpServletResponse response) throws Exception {
    System.out.println("modifyGoodsImageInfo");
    multipartRequest.setCharacterEncoding("utf-8");
    response.setContentType("text/html; charset=utf-8");
    String imageName=null;

    Map goodsMap = new HashMap();
    Enumeration enu=multipartRequest.getParameterNames();
    while(enu.hasMoreElements()){
        String name=(String)enu.nextElement();
        String value=multipartRequest.getParameter(name);
        goodsMap.put(name,value);
    }

    HttpSession session = multipartRequest.getSession();
    MemberVO memberVO = (MemberVO) session.getAttribute("memberInfo");
    String reg_id = memberVO.getMember_id();

    List<ImageFileVO> fileList=null;
    int goods_id=0;
    int image_id=0;
    try {
        fileList =upload(multipartRequest);
        if(fileList!= null && fileList.size() !=0) {
            for(ImageFileVO imageView0 : fileList) {
                goods_id = Integer.parseInt((String)goodsMap.get("goods_id"));
                image_id = Integer.parseInt((String)goodsMap.get("image_id"));
                imageView0.setGoods_id(goods_id);
                imageView0.setImage_id(image_id);
                imageView0.setReg_id(reg_id);
            }
        }

        adminGoodsService.modifyGoodsImage(fileList);
        for(ImageFileVO imageView0:imageList) {
            imageName = imageView0.getFileName();
            File srcFile = new File(CURR_IMAGE_REPO_PATH+"/"+ "temp" + "/" + imageName);
            File destDir = new File(CURR_IMAGE_REPO_PATH+ "/"+goods_id);
            FileUtils.moveFileToDirectory(srcFile, destDir,true);
        }
    }
}

```

- 이미지 수정 버튼 클릭 시 동작하는 controller

# AdminGoodsControllerImpl(5)

```
@Override  
@RequestMapping(value="/removeGoodsImage.do" ,method={RequestMethod.POST})  
public void removeGoodsImage(@RequestParam("goods_id") int goods_id,  
                           @RequestParam("image_id") int image_id,  
                           @RequestParam("imageFileName") String imageFileName,  
                           HttpServletRequest request, HttpServletResponse response) throws Exception {  
  
    adminGoodsService.removeGoodsImage(image_id);  
    try{  
        File srcFile = new File(CURR_IMAGE_REPO_PATH+"/"+goods_id+"/"+imageFileName);  
        srcFile.delete();  
    }catch(Exception e) {  
        e.printStackTrace();  
    }  
}
```

- 이미지 삭제 버튼 클릭 시 동작하는 controller

# AdminGoodsServiceImpl(1)

```

@Service("adminGoodsService")
@Transactional(propagation=Propagation.REQUIRED)
public class AdminGoodsServiceImpl implements AdminGoodsService {
    @Autowired
    AdminGoodsDAO adminGoodsDAO;

    @Override// 새 상품 등록 (goods , goods_image 데이터 입력)
    public int addNewGoods(Map newGoodsMap) throws Exception{
        int goods_id = adminGoodsDAO.insertNewGoods(newGoodsMap);
        ArrayList<ImageFileVO> fileList = (ArrayList)newGoodsMap.get("imageFileList");
        for(ImageFileVO imageFileVO : fileList) {
            imageFileVO.setGoods_id(goods_id);
        }
        adminGoodsDAO.insertGoodsImageFile(fileList);
        return goods_id;
    }

    @Override//등록한 상품 조회
    public List<GoodsVO> listNewGoods(Map condMap) throws Exception{
        return adminGoodsDAO.selectNewGoodsList(condMap);
    }
    @Override //상품수정페이지에 나오는 품목 상세 조회
    public Map goodsDetail(Map condMap) throws Exception {
        Map goodsMap = new HashMap();
        GoodsVO goodsVO=adminGoodsDAO.selectGoodsDetail(condMap);
        List fileList = adminGoodsDAO.selectGoodsImageFileList((int)condMap.get("goods_id"));
        goodsMap.put("goods",...goodsVO);
        goodsMap.put("imageFileList",...fileList);
        return goodsMap;
    }
    @Override //상품 이미지 조회
    public List goodsImageFile(int goods_id) throws Exception{
        List fileList = adminGoodsDAO.selectGoodsImageFileList(goods_id);
        return fileList;
    }

    @Override// 상품 정보 수정
    public void modifyGoodsInfo(Map goodsMap) throws Exception{
        adminGoodsDAO.updateGoodsInfo(goodsMap);
    }
}

```

adminGoodsController에서 호출 하는 비지니스 영역

# AdminGoodsServiceImpl(2)

```
@Override//상품이미지 수정
public void modifyGoodsImage(List<ImageFileVO> fileList) throws Exception{
    adminGoodsDAO.updateGoodsImage(fileList);
}

@Override//주문 상품 조회
public List<OrderVO> listOrderGoods(Map condMap) throws Exception{
    return adminGoodsDAO.selectOrderGoodsList(condMap);
}

@Override
public void modifyOrderGoods(Map orderMap) throws Exception{
    adminGoodsDAO.updateOrderGoods(orderMap);
}

@Override//상품이미지 삭제
public void removeGoodsImage(int image_id) throws Exception{
    adminGoodsDAO.deleteGoodsImage(image_id);
}

@Override//이미지 추가
public void addNewGoodsImage(List fileList) throws Exception{
    adminGoodsDAO.insertGoodsImageFile(fileList);
}

@Override//상품이미지 조회
public void addNewGoodsDetail(Map detailMap) throws Exception {
    adminGoodsDAO.insertNewGoodsDetail(detailMap);
}
    • master 테이블(goods) 이력 후 동작

@Override//goods_detail테이블 수정(상품 사이즈)
public void modifyGoodsDetailInfo(Map goodsMap) throws Exception {
    adminGoodsDAO.updateGoodsDetailInfo(goodsMap);
}
```

# AdminGoodsDAO(1)

```

@Repository("adminGoodsDAO")
public class AdminGoodsDAOImpl implements AdminGoodsDAO{
    @Autowired
    private SqlSession sqlSession;

    @Override//상품 정보 입력
    public int insertNewGoods(Map newGoodsMap) throws DataAccessException {
        sqlSession.insert("mapper.admin.goods.insertNewGoods",newGoodsMap);
        return Integer.parseInt((String)newGoodsMap.get("goods_id"));
    }

    @Override//상품이미지 입력(goods_Image 테이블)
    public void insertGoodsImageFile(List fileList) throws DataAccessException {
        for(int i=0; i<fileList.size();i++){
            ImageFileVO imageFileVO=(ImageFileVO)fileList.get(i);
            sqlSession.insert("mapper.admin.goods.insertGoodsImageFile",imageFileVO);
        }
    }

    @Override//상품 리스트 조회
    public List<GoodsVO>selectNewGoodsList(Map condMap) throws DataAccessException {
        ArrayList<GoodsVO> goodsList=(ArrayList)sqlSession.selectList("mapper.admin.goods.selectNewGoodsList",condMap);
        return goodsList;
    }

    @Override// 상품 상세 조회
    public GoodsVO selectGoodsDetail(Map condMap) throws DataAccessException{
        GoodsVO goodsBean = new GoodsVO();
        goodsBean=(GoodsVO)sqlSession.selectOne("mapper.admin.goods.selectGoodsDetail",condMap);
        return goodsBean;
    }

    @Override// 상품이 이미지 리스트 조회
    public List selectGoodsImageFileList(int goods_id) throws DataAccessException {
        List imageList=new ArrayList();
        imageList=(List)sqlSession.selectList("mapper.admin.goods.selectGoodsImageFileList",goods_id);
        return imageList;
    }

    @Override// 상품(goods 테이블) 정보수정
    public void updateGoodsInfo(Map goodsMap) throws DataAccessException{
        sqlSession.update("mapper.admin.goods.updateGoodsInfo",goodsMap);
    }
}

```

DB(DataBase)를 처리하는 persistence 영역 로직

# AdminGoodsDAO(2)

```

@Override// 상품(goods 테이블) 정보수정
public void updateGoodsInfo(Map goodsMap) throws DataAccessException{
    sqlSession.update("mapper.admin.goods.updateGoodsInfo",goodsMap);
}

@Override// 상품 이미지 삭제
public void deleteGoodsImage(int image_id) throws DataAccessException{
    sqlSession.delete("mapper.admin.goods.deleteGoodsImage",image_id);
}

@Override// 상품 이미지 리스트 삭제
public void deleteGoodsImage(List fileList) throws DataAccessException{
    int image_id;
    for(int i=0; i<fileList.size();i++){
        ImageFileVO bean=(ImageFileVO) fileList.get(i);
        image_id=bean.getImage_id();
        sqlSession.delete("mapper.admin.goods.deleteGoodsImage",image_id);
    }
}

@Override //상품 이미지 수정
public void updateGoodsImage(List<ImageFileVO> imageFileDialog) throws DataAccessException {

    for(int i=0; i<imageFileDialog.size();i++){
        ImageFileVO imageFileVO = imageFileDialog.get(i);
        sqlSession.update("mapper.admin.goods.updateGoodsImage",imageFileVO);
    }
}

@Override //goods_detail 테이블(상세 정보 :색상, 사이즈) 저장
public void insertNewGoodsDetail(Map detailMap) throws DataAccessException {
    sqlSession.insert("mapper.admin.goods.insertNewGoodsDetail",detailMap);
}

@Override//goods_detail 테이블(상세 정보 :색상, 사이즈) 수정
public void updateGoodsDetailInfo(Map goodsMap) throws DataAccessException {
    sqlSession.update("mapper.admin.goods.updateGoodsDetailInfo",goodsMap);
}

```

DB(DataBase)를 처리하는 persistence영역 로직

# mapper\_admin\_goods(1)

```

<mapper namespace="mapper.admin.goods">
    <!-- ResultMap 정의 -->
    <!-- GoodsVO, GOODS Table 매핑-->
    <resultMap type="goodsVO" id="goodsResult">
        <id property="goods_id" column="goods_id" />
        <result property="goods_sort" column="goods_sort" />
        <result property="goods_title" column="goods_title" />
        <result property="goods_price" column="goods_price" />
        <result property="goods_sales_price" column="goods_sales_price" />
        <result property="goods_manufacturer" column="goods_manufacturer"/>
        <result property="goods_production_date" column="goods_production_date" />
        <result property="goods_delivery_price" column="goods_delivery_price" />
        <result property="goods_delivery_date" column="goods_delivery_date" />
        <result property="goods_status" column="goods_status" />
        <result property="goods_intro" column="goods_intro" />
        <result property="goods_fileName" column="filename" />
        <result property="goods_size" column="goods_size" />
        <result property="goods_color" column="goods_color" />
        <result property="goods_qty" column="goods_qty" />
    </resultMap>
    <!-- ImageFileVO, GOODS_IMAGE Table 매핑-->
    <resultMap type="imageFileVO" id="imageResult">
        <id property="image_id" column="image_id" />
        <result property="goods_id" column="goods_id" />
        <result property="fileName" column="filename" />
        <result property="reg_id" column="reg_id" />
        <result property="fileType" column="filetype" />
        <!-- <result property="reg_date" column="reg_date" /> -->
    </resultMap>

```

```

<!-- OrderVO, ORDERS Table 매핑 -->
<resultMap id="orderGoodsResult" type="orderVO">
    <result property="order_id" column="order_id" />
    <result property="member_id" column="member_id" />
    <result property="goods_id" column="goods_id" />
    <result property="goods_title" column="goods_title" />
    <result property="goods_sales_price" column="goods_sales_price" />
    <!-- <result property="order_total_price" column="order_total_price" /> -->
    <result property="order_goods_qty" column="order_goods_qty" />

    <result property="orderer_name" column="orderer_name" />
    <result property="receiver_name" column="receiver_name" />
    <result property="receiver_hp1" column="receiver_hp1" />
    <result property="receiver_hp2" column="receiver_hp2" />
    <result property="receiver_hp3" column="receiver_hp3" />

    <result property="receiver_tel1" column="receiver_tel1" />
    <result property="receiver_tel2" column="receiver_tel2" />
    <result property="receiver_tel3" column="receiver_tel3" />
    <result property="delivery_address" column="delivery_address" />
    <result property="delivery_message" column="delivery_message" />

    <result property="delivery_method" column="delivery_method" />
    <result property="gift_wrapping" column="gift_wrapping" />
    <result property="pay_method" column="pay_method" />
    <result property="card_com_name" column="card_com_name" />
    <result property="card_pay_month" column="card_pay_month" />

    <result property="pay_orderer_hp" column="pay_orderer_hp" />
    <result property="pay_order_time" column="pay_order_time" />
    <result property="delivery_state" column="delivery_state" />
    <result property="final_total_price" column="final_total_price" />
    <result property="goods_qty" column="goods_qty" />
    <result property="orderer_hp" column="orderer_hp" />
    <result property="goods_fileName" column="goods_fileName" />
</resultMap>

```

관리자 상품 관리 페이지에서 동작하는 sql

# mapper\_admin\_goods(2)

```

<!-- GOODS Table 상품 등록, 일련 번호 시퀀스를 활용, 미리 생성 해서 넣어 줌 -->
<insert id="insertNewGoods" parameterType="java.util.Map" >
    <selectKey resultType="String" keyProperty="goods_id" order="BEFORE">
        SELECT SEQ_GOODS_ID.NEXTVAL FROM DUAL
    </selectKey>
    <![CDATA[
        INSERT INTO GOODS(GOODS_ID
            ,GOODS_SORT
            ,GOODS_TITLE
            ,GOODS_MANUFACTURER
            ,GOODS_PRICE
            ,GOODS_SALES_PRICE
            ,GOODS_POINT
            ,GOODS_PRODUCTION_DATE
            ,GOODS_DELIVERY_PRICE
            ,GOODS_DELIVERY_DATE
            ,GOODS_STATUS
            ,GOODS_INTRO)
        VALUES(#{goods_id}
            ,#{goods_sort}
            ,#{goods_title}
            ,#{goods_manufacturer}
            ,#{goods_price}
            ,#{goods_sales_price}
            ,#{goods_point}
            ,#{goods_production_date}
            ,#{goods_delivery_price}
            ,#{goods_delivery_date}
            ,#{goods_status}
            ,#{goods_intro})
    ]]>
</insert>

```

- selectkey : goods\_id 에 시퀀스 입력하기 위해 insert 전에 시퀀스 생성

```

<insert id="insertNewGoodsDetil" parameterType="java.util.HashMap" >
    <!-- <![CDATA[ -->
    <foreach collection="detailMap" item="item" index="index" open="INSERT ALL " separator=" " close="SELECT * FROM DUAL" >
        INTO GOODS_DETAIL(GOODS_ID
            ,GOODS_SIZE
            ,GOODS_COLOR
            ,GOODS_QTY)
        VALUES
        (
            #{item.goods_id, jdbcType=NUMERIC}
            ,#{item.goods_size, jdbcType=VARCHAR}
            ,#{item.goods_color, jdbcType=VARCHAR}
            ,#{item.goods_qty, jdbcType=NUMERIC}
        )
    </foreach>
    <!-- ]]> -->
</insert>

<!-- GOODS_IMAGE Table 상품이미지 등록 , 이미지 별로 시퀀스 생성 -->
<insert id="insertGoodsImageFile" parameterType="ImageFileVO">
    <selectKey resultType="int" keyProperty="image_id" order="BEFORE">
        SELECT SEQ_IMAGE_ID.NEXTVAL FROM DUAL
    </selectKey>
    <![CDATA[
        INSERT INTO GOODS_IMAGE(IMAGE_ID
            ,GOODS_ID
            ,FILENAME
            ,FILETYPE
            ,REG_ID)
        VALUES(#{image_id}
            ,#{goods_id}
            ,#{fileName}
            ,#{fileType}
            ,#{reg_id})
    ]]>

```

- 리스트에 있는 goods\_detail에 넣을 정보를 넣기 위해 foreach를 사용해서 사이즈별로 반복해서 입력

관리자 상품 관리 페이지에서 동작하는 sql

# mapper\_admin\_goods(3)

```
<!-- 상품 확인 페이지 (페이지 처리, 상품 입고일자로 검색) -->
<select id="selectNewGoodsList" resultMap="goodsResult" parameterType="java.util.Map">

    SELECT * FROM (SELECT ROWNUM AS RECNUM
        , t.*
        FROM (SELECT a.GOODS_ID
            , a.GOODS_TITLE
            , a.GOODS_MANUFACTURER
            , a.GOODS_SALES_PRICE
            , TO_CHAR(a.GOODS_ENTER_DATE, 'YYYY-MM-DD') AS GOODS_ENTER_DATE
            , TO_CHAR(a.GOODS_PRODUCTION_DATE, 'YYYY-MM-DD') AS GOODS_PRODUCTION_DATE
            , b.goods_size
            , b.goods_qty
            , b.goods_color
        FROM GOODS a, goods_detail b
        where a.goods_id = b.goods_id
        and TO_CHAR(a.GOODS_ENTER_DATE, 'YYYY-MM-DD') BETWEEN #{beginDate} and #{endDate}
        <if test="search_type=='goods_title'">
            AND a.GOODS_TITLE LIKE '%'||#{search_word}||'%'
        </if>
        <if test="search_type=='goods_id'">
            AND a.GOODS_ID LIKE '%'||#{search_word}||'%'
        </if>
        <if test="search_type=='goods_publisher'">
            AND a.GOODS_MANUFACTURER LIKE '%'||#{search_word}||'%'
        </if>
    ORDER BY a.GOODS_ENTER_DATE DESC)t)
    WHERE RECNUM BETWEEN (#{section}-1)*100 + (#{pageNum}-1)*10+1 AND (#{section}-1)*100 + (#{pageNum})*10
</select>
```

- 페이지 처리 하기 위해 ROWNUM 함수를 사용해서 조회
- 검색 키워드에 따라 where 조건을 변경해서 조회

관리자 상품 관리 페이지에서 동작하는 sql

# mapper\_admin\_goods(4)

```

<update id="updateGoodsInfo" parameterType="java.util.HashMap">
    UPDATE GOODS
    <set>
        <if test=" goods_sort!='' and goods_sort!=null">
            GOODS_SORT=#{goods_sort},
        </if>
        <if test=" goods_title!='' and goods_title!=null">
            GOODS_TITLE=#{goods_title},
        </if>
        <if test=" goods_manufacturer!='' and goods_manufacturer!=null">
            GOODS_MANUFACTURER=#{goods_manufacturer},
        </if>
        <if test=" goods_price!='' and goods_price!=null">
            GOODS_PRICE=#{goods_price},
        </if>
        <if test=" goods_production_date!='' and goods_production_date!=null">
            GOODS_PRODUCTION_DATE=#{goods_production_date},
        </if>
        <if test=" goods_sales_price!='' and goods_sales_price!=null">
            GOODS_SALES_PRICE=#{goods_sales_price},
        </if>
        <if test=" goods_point!='' and goods_point!=null">
            GOODS_POINT=#{goods_point},
        </if>
        <if test=" goods_delivery_price!='' and goods_delivery_price!=null">
            GOODS_DELIVERY_PRICE=#{goods_delivery_price},
        </if>
        <if test=" goods_delivery_date!='' and goods_delivery_date!=null">
            GOODS_DELIVERY_DATE=#{goods_delivery_date},
        </if>
        <if test=" goods_status!='' and goods_status!=null">
            GOODS_STATUS=#{goods_status},
        </if>
        <if test=" goods_intro!='' and goods_intro!=null">
            GOODS_INTRO=#{goods_intro},
        </if>
    </set>
    WHERE GOODS_ID = #{goods_id}
</update>

```

- 각 항목의 키값을 받아 항목 별로 수정

```

<update id="updateGoodsDetailInfo" parameterType="java.util.HashMap">
    UPDATE GOODS_DETAIL
    <set>
        <if test=" goods_qty!='' and goods_qty!=null">
            GOODS_QTY=#{goods_qty},
        </if>
    </set>
    WHERE GOODS_ID = #{goods_id}
    AND goods_size = #{goods_size}
    and goods_color = #{goods_color}
</update>

<!-- 상품이미지 테이블의 상품 이미지 수정 -->
<update id="updateGoodsImage" parameterType="imageFileVO">
    <![CDATA[
        UPDATE GOODS_IMAGE
        SET FILENAME=#{fileName}
        WHERE GOODS_ID=#{goods_id}
        AND IMAGE_ID=#{image_id}
    ]]>
</update>

<!-- 상품이미지 테이블에서 상품 이미지를 삭제 -->
<delete id="deleteGoodsImage" parameterType="String">
    <![CDATA[
        DELETE FORM GOODS_IMAGE
        WHERE IMAGE_ID=#{image_id}
    ]]>
</delete>

```

# adminOrderController(1)

```

@Controller("adminOrderController")
@RequestMapping(value="/admin/order")
public class AdminOrderControllerImpl extends BaseController implements AdminOrderController{
    @Autowired
    AdminOrderService adminOrderService;

    @RequestMapping(value="/adminOrderMain.do" ,method={RequestMethod.GET, RequestMethod.POST})
    public ModelAndView adminOrderMain(@RequestParam Map<String, String> dateMap,
                                       HttpServletRequest request, HttpServletResponse response) throws Exception {
        String viewName=(String)request.getAttribute("viewName");
        ModelAndView mav = new ModelAndView(viewName);

        String fixedSearchPeriod = dateMap.get("fixedSearchPeriod");
        String section = dateMap.get("section");
        String pageNum = dateMap.get("pageNum");
        String search_type = dateMap.get("search_type");
        String search_word = dateMap.get("search_word");
        String beginDate=dateMap.get("beginDate");
        String endDate=dateMap.get("endDate");

        if(beginDate == null) {
            String [] tempDate=calcSearchPeriod(fixedSearchPeriod).split(",");
            beginDate=tempDate[0];
            endDate=tempDate[1];
        }

        HashMap<String, Object> condMap=new HashMap<String, Object>();
        if(section== null) {
            section = "1";
        }
        condMap.put("section", section);
        if(pageNum== null) {
            pageNum = "1";
        }
        condMap.put("pageNum", pageNum);
        condMap.put("beginDate", beginDate);
        condMap.put("endDate", endDate);
        condMap.put("search_type", search_type);
        condMap.put("search_word", search_word);
        List<OrderVO> newOrderList=adminOrderService.listNewOrder(condMap);
        mav.addObject("newOrderList", newOrderList);
    }
}

```

- 지정된 날짜 별, 키워드 검색으로 조회

```

String beginDate1[] = beginDate.split("-");
String endDate2[] = endDate.split("-");
mav.addObject("beginYear", beginDate1[0]);
mav.addObject("beginMonth", beginDate1[1]);
mav.addObject("beginDay", beginDate1[2]);
mav.addObject("endYear", endDate2[0]);
mav.addObject("endMonth", endDate2[1]);
mav.addObject("endDay", endDate2[2]);
mav.addObject("section", section);
mav.addObject("pageNum", pageNum);
return mav;
}

```

# adminOrderController(2)

```

@RequestMapping(value="/modifyDeliveryState.do" ,method={RequestMethod.POST})
public ResponseEntity modifyDeliveryState(@RequestParam Map<String, String> deliveryMap,
                                         HttpServletRequest request, HttpServletResponse response) throws Exception {
    adminOrderService.modifyDeliveryState(deliveryMap);

    String message = null;
    ResponseEntity resEntity = null;
    HttpHeaders responseHeaders = new HttpHeaders();
    message = "mod_success";
    resEntity =new ResponseEntity(message, responseHeaders, HttpStatus.OK);
    return resEntity;
}

@Override
@RequestMapping(value="/orderDetail.do" ,method={RequestMethod.GET,RequestMethod.POST})
public ModelAndView orderDetail(@RequestParam("order_id") int order_id,
                                 HttpServletRequest request, HttpServletResponse response) throws Exception {
    String viewName=(String)request.getAttribute("viewName");
    ModelAndView mav = new ModelAndView(viewName);
    Map orderMap =adminOrderService.orderDetail(order_id);
    mav.addObject("orderMap", orderMap);
    return mav;
}

```

- 주문 상태 변경 버튼 클릭 시 호출 되는 controller
- 주문 상세페이지의 변경 버튼에서도 함께 사용

- 주문번호 클릭 시 호출되는 controller
- 상세화면으로 이동

관리자 주문 메인화면 Controller

```
@Service("adminOrderService")
@Transactional(propagation= Propagation.REQUIRED)
public class AdminOrderServiceImpl implements AdminOrderService{

    @Autowired
    AdminOrderDAO adminOrderDAO;

    @Override
    public List<OrderVO> listNewOrder(Map condMap) throws Exception {
        return adminOrderDAO.selectNewOrderList(condMap);
    }

    @Override
    public void modifyDeliveryState(Map deliveryMap) throws Exception {
        adminOrderDAO.updateDeliveryState(deliveryMap);
    }

    @Override
    public Map<String, Object> orderDetail(int order_id) throws Exception {
        Map<String, Object> orderMap = new HashMap();
        ArrayList<OrderVO> orderList = adminOrderDAO.selectOrderDetail(order_id);
        OrderVO deliveryInfo = (OrderVO) orderList.get(0);
        String member_id = (String)deliveryInfo.getMember_id();
        MemberVO orderer = adminOrderDAO.selectOrderer(member_id);
        orderMap.put("orderList", orderList);
        orderMap.put("deliveryInfo", deliveryInfo);
        orderMap.put("orderer", orderer);

        return orderMap;
    }
}
```

adminGoodsController에서 호출 하는 비지니스 영역

```
5 @Repository("adminOrderDAO")
6 public class AdminOrderDAOImpl implements AdminOrderDAO{
7
8     @Autowired
9     private SqlSession sqlSession;
10
11     @Override 주문 목록
12     public ArrayList<OrderVO> selectNewOrderList(Map<String, Object> condMap) throws DataAccessException {
13         ArrayList<OrderVO> orderList = (ArrayList<OrderVO>) sqlSession.selectList("mapper.admin.order.selectNewOrderList", condMap);
14         return orderList;
15     }
16
17     @Override 주문 상태 변경
18     public void updateDeliveryState(Map<String, Object> deliveryMap) throws DataAccessException {
19         sqlSession.update("mapper.admin.order.updateDeliveryState", deliveryMap);
20     }
21
22     @Override 주문 정보 조회
23     public ArrayList<OrderVO> selectOrderDetail(int order_id) throws DataAccessException {
24         ArrayList<OrderVO> orderList = (ArrayList<OrderVO>) sqlSession.selectList("mapper.admin.order.selectOrderDetail", order_id);
25         return orderList;
26     }
27     주문자 정보 조회
28     @Override
29     public MemberVO selectOrderer(String member_id) throws DataAccessException {
30         MemberVO orderer = (MemberVO) sqlSession.selectOne("mapper.admin.order.selectOrderer", member_id);
31         return orderer;
32     }
33
34 }
```

adminGoodsService에서 호출 하는 persistence 영역

# mapper\_admin\_order(1)

```
<mapper namespace="mapper.admin.order">
<!!-- 리절트 맵 정의 -->
<resultMap id="orderResult" type="orderVO">
    <!-- <result property="order_seq_num" column="order_seq_num" /> -->
    <result property="order_id" column="order_id" />
    <result property="member_id" column="member_id" />
    <result property="goods_id" column="goods_id" />
    <result property="goods_title" column="goods_title" />
    <result property="goods_sales_price" column="goods_sales_price" />
    <!-- <result property="order_total_price" column="order_total_price" />
    <result property="order_goods_qty" column="order_goods_qty" />

    <result property="orderer_name" column="orderer_name" />
    <result property="receiver_name" column="receiver_name" />
    <result property="receiver_hp1" column="receiver_hp1" />
    <result property="receiver_hp2" column="receiver_hp2" />
    <result property="receiver_hp3" column="receiver_hp3" />

    <result property="receiver_tel1" column="receiver_tel1" />
    <result property="receiver_tel2" column="receiver_tel2" />
    <result property="receiver_tel3" column="receiver_tel3" />
    <result property="delivery_address" column="delivery_address" />
    <result property="delivery_message" column="delivery_message" />

    <result property="delivery_method" column="delivery_method" />
    <result property="gift_wrapping" column="gift_wrapping" />
    <result property="pay_method" column="pay_method" />
    <result property="card_com_name" column="card_com_name" />
    <result property="card_pay_month" column="card_pay_month" />

    <result property="pay_orderer_hp" column="pay_orderer_hp" />
    <result property="pay_order_time" column="pay_order_time" />
    <result property="delivery_state" column="delivery_state" />
    <result property="final_total_price" column="final_total_price" />
    <result property="goods_qty" column="goods_qty" />
    <!-- <result property="orderer_hp" column="orderer_hp" /> -->
    <result property="goods_fileName" column="goods_fileName" />
</resultMap>
```

order, member table의 리절트 맵 정의

```
<resultMap id="memberResult" type="MemberVO">
    <result property="member_id" column="member_id" />
    <result property="member_pw" column="member_pw" />
    <result property="member_name" column="member_name" />
    <result property="member_gender" column="member_gender" />

    <result property="birth_year" column="birth_year" />
    <result property="birth_month" column="birth_month" />
    <result property="birth_day" column="birth_day" />

    <result property="tel1" column="tel1" />
    <result property="tel2" column="tel2" />
    <result property="tel3" column="tel3" />

    <result property="hp1" column="hp1" />
    <result property="hp2" column="hp2" />
    <result property="hp3" column="hp3" />
    <result property="sms_yn" column="sms_yn" />
    <result property="email1" column="email1" />
    <result property="email2" column="email2" />
    <result property="email_yn" column="email_yn" />

    <result property="zipcode" column="zipcode" />
    <result property="road_address" column="road_address" />
    <result property="jibun_address" column="jibun_address" />
    <result property="detail_address" column="detail_address" />
    <result property="join_date" column="join_date" />
</resultMap>
```

# mapper\_admin\_order(2)

```

<!-- 주문 전체 보기, 페이지 처리(한 페이지 10건, 한 섹션에 10페이지) -->
<select id="selectNewOrderList" resultMap="orderResult" parameterType="orderVO" >
    SELECT * FROM (SELECT ROWNUM AS RECNUM
                  , T.*
                  FROM (SELECT *
                         FROM ORDERS
                         WHERE TO_CHAR(PAY_ORDER_TIME, 'YYYY-MM-DD') BETWEEN #{beginDate} and #{endDate}

                         <if test="search_type=='orderer_name'">
                             AND ORDERER_NAME LIKE '%'||#{search_word}||'%'
                         </if>
                         <if test="search_type=='orderer_id'">
                             AND MEMBER_ID LIKE '%'||#{search_word}||'%'
                         </if>
                         <if test="search_type=='orderer_hp'">
                             AND ORDERER_HP LIKE '%'||#{search_word}||'%'
                         </if>
                         <if test="search_type=='orderer_goods'">
                             AND GOODS_TITLE LIKE '%'||#{search_word}||'%'
                         </if>
                         ORDER BY PAY_ORDER_TIME DESC
                     ) T)
    WHERE RECNUM BETWEEN (#[section]-1)*100 + (#[pageNum]-1)*10+1 and (#[section]-1)*100 + (#[pageNum])*10
</select>

<!-- 주문 상태 변경 -->
<update id="updateDeliveryState" parameterType="java.util.Map">
    UPDATE ORDERS
    <set>
        <if test=" delivery_state=='' and delivery_state!=null">
            DELIVERY_STATE=#{delivery_state},
        </if>
    </set>
    WHERE ORDER_ID = #[order_id]
</update>

```

주문 전체 조회, 주문 상태 변경 하는 sql

## mapper\_admin\_order(3)

```
<!-- 주문 상세 보기 -->
<select id="selectOrderDetail" resultMap="orderResult"    parameterType="int" >
    <![CDATA[
        SELECT *
        FROM ORDERS
        WHERE ORDER_ID=#{order_id}
    ]]>
</select>

<!-- 회원 정보 보기 -->
<select id="selectOrderer" resultMap="memberResult"    parameterType="String" >
    <![CDATA[
        SELECT *
        FROM MEMBER
        WHERE MEMBER_ID=#{member_id}
    ]]>
</select>
```

주문 상세 조회, 주문회원 정보 조회 하는 sql

# adminMemberController(1)

```

@Controller("adminMemberController")
@RequestMapping(value="/admin/member")
public class AdminMemberControllerImpl extends BaseController implements AdminMemberController{
    @Autowired
    AdminMemberService adminMemberService;

    @RequestMapping(value="/adminMemberMain.do" ,method={RequestMethod.POST,RequestMethod.GET})
    public ModelAndView adminMemberMain(@RequestParam Map<String, String> dateMap,
                                        HttpServletRequest request,
                                        HttpServletResponse response) throws Exception{
        String viewName=(String)request.getAttribute("viewName");
        ModelAndView mav = new ModelAndView(viewName);

        String fixedSearchPeriod = dateMap.get("fixedSearchPeriod");
        String section = dateMap.get("section");
        String pageNum = dateMap.get("pageNum");
        String search_type = dateMap.get("search_type");
        String search_word = dateMap.get("search_word");
        String beginDate = dateMap.get("beginDate");
        String endDate=dateMap.get("endDate");
        System.out.println(beginDate);
        System.out.println(endDate);
        if(beginDate == null) {
            String [] tempDate=calcSearchPeriod(fixedSearchPeriod).split(" ");
            beginDate=tempDate[0];
            endDate=tempDate[1];
        }
        dateMap.put("beginDate", beginDate);
        dateMap.put("endDate", endDate);

        HashMap<String, Object> condMap=new HashMap<String, Object>();
        if(section== null) {
            section = "1";
        }
        condMap.put("section",section);
        if(pageNum== null) {
            pageNum = "1";
        }
        condMap.put("pageNum",pageNum);
        condMap.put("beginDate",beginDate);
    }
}

```

```

HashMap<String, Object> condMap=new HashMap<String, Object>();
if(section== null) {
    section = "1";
}
condMap.put("section",section);
if(pageNum== null) {
    pageNum = "1";
}
condMap.put("pageNum",pageNum);
condMap.put("beginDate",beginDate);
condMap.put("endDate", endDate);
condMap.put("search_type",search_type);
condMap.put("search_word", search_word);

ArrayList<MemberVO> member_list=adminMemberService.listMember(condMap);
mav.addObject("member_list", member_list);

String beginDate1[]=beginDate.split("-");
String endDate2[]=endDate.split("-");
mav.addObject("beginYear",beginDate1[0]);
mav.addObject("beginMonth",beginDate1[1]);
mav.addObject("beginDay",beginDate1[2]);
mav.addObject("endYear",endDate2[0]);
mav.addObject("endMonth",endDate2[1]);
mav.addObject("endDay",endDate2[2]);

mav.addObject("section", section);
mav.addObject("pageNum", pageNum);
return mav;
}

```

관리자 회원관리 메인 페이지 호출 시 동작하는 controller

```

@RequestMapping(value="/modifyMemberInfo.do" ,method={RequestMethod.POST,RequestMethod.GET})
public void modifyMemberInfo(HttpServletRequest request, HttpServletResponse response) throws Exception{
    HashMap<String, String> memberMap=new HashMap<String, String>();
    String val[] = null;
    PrintWriter pw=response.getWriter();
    String member_id=request.getParameter("member_id");
    String mod_type=request.getParameter("mod_type");
    String value =request.getParameter("value");
    if(mod_type.equals("member_birth")){
        val=value.split(",");
        memberMap.put("birth_year",val[0]);
        memberMap.put("birth_month",val[1]);
        memberMap.put("birth_day",val[2]);
    }else if(mod_type.equals("tel")){
        val=value.split(",");
        memberMap.put("tel1",val[0]);
        memberMap.put("tel2",val[1]);
        memberMap.put("tel3",val[2]);
    }else if(mod_type.equals("hp")){
        val=value.split(",");
        memberMap.put("hp1",val[0]);
        memberMap.put("hp2",val[1]);
        memberMap.put("hp3",val[2]);
        memberMap.put("sms_yn", val[3]);
    }else if(mod_type.equals("email")){
        val=value.split(",");
        memberMap.put("email1",val[0]);
        memberMap.put("email2",val[1]);
        memberMap.put("email_yn", val[2]);
    }else if(mod_type.equals("address")){
        val=value.split(",");
        memberMap.put("zipcode",val[0]);
        memberMap.put("road_address",val[1]);
        memberMap.put("jibun_address", val[2]);
        memberMap.put("detail_address", val[3]);
    } else if(mod_type.equals("member_pw")){
        memberMap.put("member_pw", value);
    }else if(mod_type.equals("member_gender")){
        memberMap.put("member_gender", value);
    }
}

```

변경 할 멤버 정보의 키값에 따라 한 항목 수정

memberMap.put("member\_id", member\_id);  
 adminMemberService.modifyMemberInfo(memberMap);  
 pw.print("mod\_success");  
 pw.close();

관리자 회원관리 상세 페이지에서 수정 버튼 클릭 시 동작 하는 controller

## adminMemberController(3)

```

@RequestMapping(value="/memberDetail.do" ,method={RequestMethod.POST,RequestMethod.GET})
public ModelAndView memberDetail(HttpServletRequest request, HttpServletResponse response) throws Exception{
    String viewName=(String)request.getAttribute("viewName");
    ModelAndView mav = new ModelAndView(viewName);          회원 관리 메인페이지에서 회원 아이디 클릭 시 호출
    String member_id=request.getParameter("member_id");
    MemberVO member_info=adminMemberService.memberDetail(member_id);
    mav.addObject("member_info",member_info);
    return mav;
}          회원 탈퇴 버튼 클릭 시 호출

@RequestMapping(value="/deleteMember.do" ,method={RequestMethod.POST})
public ModelAndView deleteMember(HttpServletRequest request, HttpServletResponse response) throws Exception {
    ModelAndView mav = new ModelAndView();
    HashMap<String, String> memberMap=new HashMap<String, String>();
    String member_id=request.getParameter("member_id");
    String delete_yn=request.getParameter("delete_yn");
    memberMap.put("delete_yn", delete_yn);
    memberMap.put("member_id", member_id);

    adminMemberService.modifyMemberInfo(memberMap);
    mav.setViewName("redirect:/admin/member/adminMemberMain.do");
    return mav;
}

```

관리자 회원관리 상세 페이지에서 수정 버튼 클릭 시 동작 하는 controller

# adminMemberService

```
@Service("adminMemberService")
@Transactional(propagation=Propagation.REQUIRED)
public class AdminMemberServiceImpl implements AdminMemberService {
    @Autowired
    AdminMemberDAO adminMemberDAO;
    public ArrayList<MemberVO> listMember(HashMap condMap) throws Exception{
        return adminMemberDAO.listMember(condMap);
    }

    public MemberVO memberDetail(String member_id) throws Exception{
        return adminMemberDAO.memberDetail(member_id);
    }

    public void modifyMemberInfo(HashMap memberMap) throws Exception{
        String member_id=(String)memberMap.get("member_id");
        adminMemberDAO.modifyMemberInfo(memberMap);
    }
}
```

adminMemberController에서 호출 하는 비지니스 영역

# mapper\_admin\_member(1)

```
<mapper namespace="mapper.admin.member">
<!-- 리절트 맵 정의 -->
<resultMap id="memberResult" type="MemberVO">
    <result property="member_id" column="member_id" />
    <result property="member_pw" column="member_pw" />
    <result property="member_name" column="member_name" />
    <result property="member_gender" column="member_gender" />

    <result property="birth_year" column="birth_year" />
    <result property="birth_month" column="birth_month" />
    <result property="birth_day" column="birth_day" />

    <result property="tel1" column="tel1" />
    <result property="tel2" column="tel2" />
    <result property="tel3" column="tel3" />

    <result property="hp1" column="hp1" />
    <result property="hp2" column="hp2" />
    <result property="hp3" column="hp3" />
    <result property="sms_yn" column="sms_yn" />
    <result property="email1" column="email1" />
    <result property="email2" column="email2" />
    <result property="email_yn" column="email_yn" />

    <result property="zipcode" column="zipcode" />
    <result property="road_address" column="road_address" />
    <result property="jibun_address" column="jibun_address" />
    <result property="detail_address" column="detail_address" />
    <result property="join_date" column="join_date" />
    <result property="delete_yn" column="delete_yn" />
</resultMap>
```

member table의 리절트 맵

# mapper\_admin\_member(2)

```

<!-- 회원 리스트 보기, 페이지 처리(한 페이지 10건, 한 섹션에 10페이지) -->
<select id="listMember" resultMap="memberResult" parameterType="java.util.HashMap" >
    SELECT * FROM (SELECT ROWNUM AS RECNUM
                  , T.*
                  FROM (SELECT *
                         FROM MEMBER
                         WHERE TO_CHAR(JOIN_DATE, 'YYYY-MM-DD') BETWEEN #{beginDate} and #{endDate}
                           <if test="search_type=='member_id'">
                               AND MEMBER_ID LIKE '%'||#{search_word}||'%'
                           </if>
                           <if test="search_type=='member_name'">
                               AND MEMBER_NAME LIKE '%'||#{search_word}||'%'
                           </if>
                           <if test="search_type=='member_hp_num'">
                               AND hp1||hp2||hp3 LIKE '%'||#{search_word}||'%'
                           </if>
                           <if test="search_type=='member_address'">
                               AND JIBUN_ADDRESS||ROAD_ADDRESS||DETAIL_ADDRESS LIKE '%'||#{search_word}||'%'
                           </if>
                         ORDER BY JOIN_DATE DESC
                      ) T)
    WHERE RECNUM BETWEEN (#{section}-1)*100 + (#{pageNum}-1)*10+1 and (#{section}-1)*100 + (#{pageNum})*10
</select>

<!-- 아이디로 회원조회 -->
<select id="memberDetail" resultMap="memberResult" parameterType="String" >
    <![CDATA[
        SELECT *
        FROM MEMBER
        WHERE MEMBER_ID=#{member_id}
    ]]>
</select>

```

회원 목록 조회, 아이디로 한명 상세 조회

# mapper\_admin\_member(3)

```

<update id="modifyMemberInfo" parameterType="java.util.HashMap">
    UPDATE MEMBER
    <set>
        <if test=" member_pw!='' and member_pw!=null">
            member_pw=#{member_pw},
        </if>
        <if test=" member_gender!='' and member_gender!=null">
            MEMBER_GENDER=#{member_gender},
        </if>
        <if test=" birth_year!='' and birth_year!=null">
            BIRTH_YEAR=#{birth_year},
        </if>
        <if test=" birth_month!='' and birth_month!=null">
            BIRTH_MONTH=#{birth_month},
        </if>
        <if test=" birth_day!='' and birth_day!=null">
            BIRTH_DAY=#{birth_day},
        </if>
        <if test=" tel1!='' and tel1!=null">
            TEL1=#{tel1},
        </if>
        <if test=" tel2!='' and tel2!=null">
            TEL2=#{tel2},
        </if>
        <if test=" tel3!='' and tel3!=null">
            TEL3=#{tel3},
        </if>
        <if test=" hp1!='' and hp1!=null">
            HP1=#{hp1},
        </if>
        <if test=" hp2!='' and hp2!=null">
            HP2=#{hp2},
        </if>
        <if test=" hp3!='' and hp3!=null">
            HP3=#{hp3},
        </if>
        <if test=" sms_yn!='' and sms_yn!=null">
            SMS_YN=#{sms_yn},
        </if>

```

```

        <if test=" email2!='' and email2!=null">
            EMAIL2=#{email2},
        </if>
        <if test=" email3!='' and email3!=null">
            EMAIL3=#{email3},
        </if>
        <if test=" email_yn!='' and email_yn!=null">
            EMAIL_YN=#{email_yn},
        </if>
        <if test=" zipcode!='' and zipcode!=null">
            ZIPCODE=#{zipcode},
        </if>
        <if test=" road_address!='' and road_address!=null">
            ROAD_ADDRESS=#{road_address},
        </if>
        <if test=" jibun_address!='' and jibun_address!=null">
            JIBUN_ADDRESS=#{jibun_address},
        </if>
        <if test=" detail_address!='' and detail_address!=null">
            DETAIL_ADDRESS=#{detail_address},
        </if>
        <if test=" delete_yn!='' and delete_yn!=null">
            DELETE_YN=#{delete_yn},
        </if>
    </set>
    WHERE
    MEMBER_ID=#{member_id}
</update>

```

컬럼명의 키값을 받아 회원 정보 수정

- Tiles Framework를 사용하여 include 형식을 편리한 유지보수를 위하여 사용
- header, body, side, footer, quick으로 나누어 설계

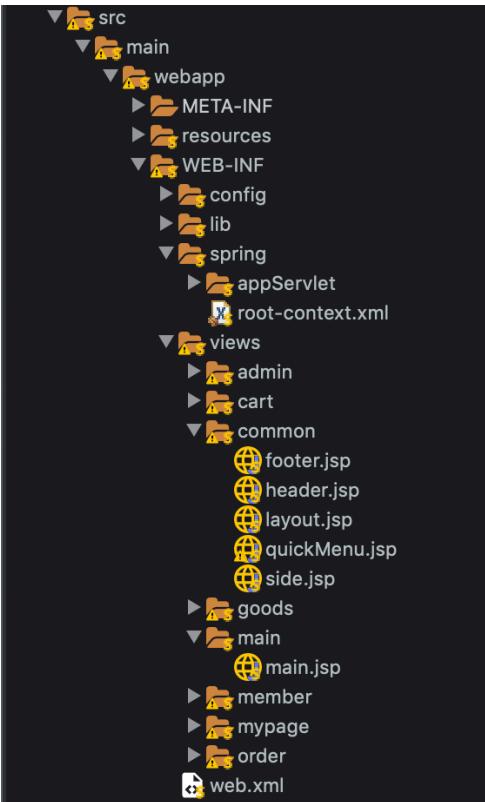
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE tiles-definitions PUBLIC
    "-//Apache Software Foundation//DTD Tiles Configuration 3.0//EN"
    "http://tiles.apache.org/dtds/tiles-config_3_0.dtd">
<tiles-definitions>
    <definition name="baseLayout" template="/WEB-INF/views/common/layout.jsp">
        <put-attribute name="title" value="레이아웃 페이지"/>
        <put-attribute name="header" value="/WEB-INF/views/common/header.jsp"/>
        <put-attribute name="side" value="/WEB-INF/views/common/side.jsp"/>
        <put-attribute name="body" value="" />
        <put-attribute name="footer" value="/WEB-INF/views/common/footer.jsp"/>
        <put-attribute name="quickMenu" value="/WEB-INF/views/common/quickMenu.jsp"/>
    </definition>
    <definition name="/main/main" extends="baseLayout">
        <put-attribute name="title" value="FootBallMall 메인페이지"/>
        <put-attribute name="body" value="/WEB-INF/views/main/main.jsp" />
    </definition>
</tiles-definitions>
```

- template : 전체 페이지의 틀
- put-attribute : 각 페이지의 요소
- value : jsp파일 경로 지정
- extends : definition의 name(controller의 url)에 일치하면 baseLayout의 body에 화면을 출력

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE tiles-definitions PUBLIC
    "-//Apache Software Foundation//DTD Tiles Configuration 3.0//EN"
    "http://tiles.apache.org/dtds/tiles-config_3_0.dtd">
<tiles-definitions>
    <definition name="baseLayout" template="/WEB-INF/views/common/layout.jsp">
        <put-attribute name="title" value="레이아웃 페이지"/>
        <put-attribute name="header" value="/WEB-INF/views/common/header.jsp"/>
        <put-attribute name="side" value="/WEB-INF/views/common/side.jsp"/>
        <put-attribute name="body" value="" />
        <put-attribute name="footer" value="/WEB-INF/views/common/footer.jsp"/>
        <put-attribute name="quickMenu" value="/WEB-INF/views/common/quickMenu.jsp"/>
    </definition>
    <definition name="/main/main" extends="baseLayout">
        <put-attribute name="title" value="FootBallMall 메인페이지"/>
        <put-attribute name="body" value="/WEB-INF/views/main/main.jsp" />
    </definition>
</tiles-definitions>
```

- template : 전체 페이지의 틀
- put-attribute : 각 페이지의 요소
- value : jsp파일 경로 지정
- extends : definition의 name(controller의 url)에 일치하면 baseLayout의 body에 화면을 출력

# layout.jsp



layout.jsp의 기본 구조

```
<title><tiles:insertAttribute name="title" /></title>

</head>
<body>
    <div id="outer_wrap">
        <div id="wrap">
            <header>
                <tiles:insertAttribute name="header" />
            </header>
            <div class="clear"></div>
            <aside>
                <tiles:insertAttribute name="side" />
            </aside>
            <article>
                <tiles:insertAttribute name="body" />
            </article>
            <div class="clear"></div>
            <footer>
                <tiles:insertAttribute name="footer" />
            </footer>
        </div>
        <tiles:insertAttribute name="quickMenu" />
    </div>
</body>
```

02

본론

# layout.jsp

header

FootBall Mall

로그인

회원가입

검색

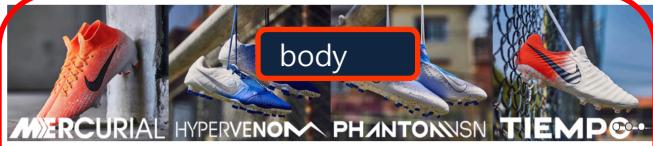
side

축구용품

- 축구화
- 유니폼 상의
- 유니폼 하의
- 크루프
- 트레이닝복
- 액세서리



축구용품계의 신 뜻볼을!!



body



quick



스테디셀러



footer



(주)풋볼  
대표이사: 외지현  
주소: 우편번호 03153 서울시 종로구 종로5  
사업자등록번호: 102-81-1111  
대표전화: 02-123-4567, 팩스: 02-123-4567  
인터넷전화: 1544-1544 (인터넷 부산전화) 팩스: 0502-977-7777 ( 지역번호별로 최고책임자 : 민지현 aj03169@gmail.com )  
대표전화: 1544-1544 (인터넷 부산전화) 팩스: 0502-977-7777 ( 지역번호별로 최고책임자 : 민지현 aj03169@gmail.com )  
COPYRIGHT © 2011. 웹사이트: FootBall All Rights Reserved.

웹상에서 보이는 화면

02

본론

# header.jsp

FootBall Mall

로그인

회원가입

검색

로그아웃 상태 header

FootBall Mall

마이페이지

장바구니

로그아웃

검색

회원 로그인 상태 header

FootBall Mall

마이페이지

장바구니

관리자

로그아웃

검색

관리자 로그인 상태 header

# header.jsp

```
8<ul class="nav navbar-nav">
9    <c:choose>
0        <c:when test="${isLogOn==true and not empty memberInfo }">
1            <li></li>
2            <li></li>
3            <li></li>
4            <li></li>
5            <li></li>
6            <li></li>
7            <li><a href="${contextPath}/mypage/myPageMain.do">마이페이지</a></li>
8            <li><a href="${contextPath}/cart/myCartList.do">장바구니</a></li>
9            <c:if test="${isLogOn==true and memberInfo.member_id =='admin' }">
0                <li class="no_line"><a href="${contextPath}/admin/goods/adminGoodsMain.do">관리자</a></li>
1            </c:if>
2        </c:when>
3    </c:choose>
4</ul>
5<form class="navbar-form navbar-right" name="frmSearch" action="${contextPath}/goods/searchGoods.do" role="search">
6    <input name="searchWord" class="form-control" type="text" onKeyUp="keywordSearch()"/>
7    <input type="submit" class="btn btn-default" name="search" class="btn1" value="검색" />
8</form>
9<div id="suggest">
0    <div id="suggestList"></div>
1</div>
2<ul class="nav navbar-nav navbar-right">
3    <c:choose>
4        <c:when test="${isLogOn==true and not empty memberInfo }">
5            <li><a href="${contextPath}/member/logout.do">로그아웃</a></li>
6        </c:when>
7        <c:otherwise>
8            <li><a href="${contextPath}/member/loginForm.do">로그인</a></li>
9            <li><a href="${contextPath}/member/memberForm.do">회원가입</a></li>
0        </c:otherwise>
1    </c:choose>
2</ul>
3</div><!-- /.navbar-collapse -->
4</div><!-- /.container-fluid -->
5</nav>
```

로그인 유/무 관계에 따라 달라지는 header

## 축구용품

- 축구화
- 유니폼 상의
- 유니폼 하의
- 크루탑
- 트레이닝 복
- 악세사리

메인화면에서 side 메뉴

## 주문내역

[주문내역/배송 조회](#)

## 정보내역

[회원정보관리](#)

마이페이지에서 side 화면

## 주요기능

- 상품관리
- 주문관리
- 회원관리
- 배송관리
- 게시판관리



관리자 페이지에서 side 화면

```

<c:choose>
    <c:when test="${side_menu=='admin_mode' }">
        <li>
            <H3>주요기능</H3>
            <ul>
                <li><a href="${contextPath}/admin/goods/adminGoodsMain.do">상품관리</a></li>
                <li><a href="${contextPath}/admin/order/adminOrderMain.do">주문관리</a></li>
                <li><a href="${contextPath}/admin/member/adminMemberMain.do">회원관리</a></li>
                <li><a href="#">배송관리</a></li>
                <li><a href="#">게시판관리</a></li>
            </ul>
        </li>
    </c:when>
    <c:when test="${side_menu=='my_page' }">
        <li>
            <h3>주문내역</h3>
            <ul>
                <li><a href="${contextPath}/mypage/listMyOrderHistory.do">주문내역/배송 조회</a></li>
            </ul>
        </li>
        <li>
            <h3>정보내역</h3>
            <ul>
                <li><a href="${contextPath}/mypage/myDetailInfo.do">회원정보관리</a></li>
            </ul>
        </li>
    </c:when>
    <c:otherwise>
        <li>
            <h3>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; 촉구용품</h3>
            <ul>
                <li><a href="${contextPath}/goods/goodsList.do?goods_sort=footballshoes">축구화</a></li>
                <li><a href="${contextPath}/goods/goodsList.do?goods_sort=uniform_top">유니폼 상의</a></li>
                <li><a href="${contextPath}/goods/goodsList.do?goods_sort=uniform_pants">유니폼 하의</a></li>
                <li><a href="${contextPath}/goods/goodsList.do?goods_sort=sweater">크루탑</a></li>
                <li><a href="${contextPath}/goods/goodsList.do?goods_sort=train">트레이닝복</a></li>
                <li><a href="${contextPath}/goods/goodsList.do?goods_sort=accessorie">악세사리</a></li>
            </ul>
        </li>
    </c:otherwise>

```

AdminGoodsControllerImpl에서 side\_menu의  
값 전달

MyPageControllerImpl에서 side\_menu의 값 전달

MainController에서 side\_menu의 값 전달

로그인 유/무 관계에 따라 달라지는 header

```
<head>
<link href="${contextPath}/resources/bootstrap-3.3.2-dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<html>
<body>
<footer class="footer">
<div class="container">
<div class="text-muted" style="padding-left:200px; background-color: black;">
    &nbsp; <br>
    (주)풋볼몰<br>
    대표이사: 안지혁 <br>
    주소 : 대구시 북구 호암로 40 <br>
    사업자등록번호 : 102-81-11111 <br>
    대구광역시 통신판매업신고번호 : 제 666호 ►사업자정보확인 개인정보보호최고책임자 : 안지혁 ajh3166@gmail.com <br>
    대표전화 : 1233-1233 (발신자 부담전화) 팩스 : 0502-123-1234 (지역번호공통) <br>
    COPYRIGHT(C) 풋볼몰 FootBall ALL RIGHTS RESERVED.
</div>
</div>
</footer>
</body>
</html>
```

화면 상 하단에 고정 되어 있는 footer

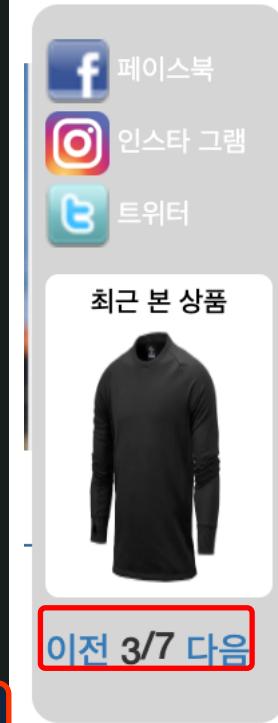
```

3      <!-- 상품이 없습니다. -->
4      <c:choose>
5          <c:when test="${ empty quickGoodsList }">
6              <strong>상품이 없습니다.</strong>
7          </c:when>
8          <c:otherwise>
9              <form name="frm_sticky" ...>
10                 <c:forEach var="item" items="${quickGoodsList}" varStatus="itemNum">
11                     <c:choose>
12                         <c:when test="${itemNum.count==1 }">
13                             <a href="javascript:goodsDetail();">
14                                 
16                             </a>
17                             <input type="hidden" name="h_goods_id" value="${item.goods_id}" />
18                             <input type="hidden" name="h_goods_fileName" value="${item.goods_fileName}" />
19                         <br>
20                     </c:when>
21                     <c:otherwise>
22                         <input type="hidden" name="h_goods_id" value="${item.goods_id}" />
23                         <input type="hidden" name="h_goods_fileName" value="${item.goods_fileName}" />
24                     </c:otherwise>
25                 </c:choose>
26             </c:forEach>
27         </c:otherwise>
28     </c:choose>
29     </ul>
30 </form>
31 </div>
32 <div>
33     <c:choose>
34         <c:when test="${ empty quickGoodsList }">
35             <h5>    0/0 </h5>
36         </c:when>
37         <c:otherwise>
38             <h5><a href='javascript:fn_show_previous_goods();'> 이전 </a> <span id="cur_goods_num">1</span>/${quickGoodsListNum}
39             <a href='javascript:fn_show_next_goods();'> 다음 </a> </h5>
40         </c:otherwise>
41     </c:choose>
42 
```

이전, 다음 버튼 클릭 시 자바스크립트로 처리

<a href='javascript:fn\_show\_previous\_goods();'> 이전 </a> <span id="cur\_goods\_num">1</span>/\${quickGoodsListNum}

<a href='javascript:fn\_show\_next\_goods();'> 다음 </a> </h5>



```

9<script>
0    var array_index=0;
1    var SERVER_URL="${contextPath}/thumbnails.do";
2    function fn_show_next_goods(){
3        var img_sticky=document.getElementById("img_sticky");
4        var cur_goods_num=document.getElementById("cur_goods_num");
5        var _h_goods_id=document.frm_sticky.h_goods_id;
6        console.log(_h_goods_id);
7        var _h_goods_fileName=document.frm_sticky.h_goods_fileName;
8        if(array_index <_h_goods_id.length-1)
9            array_index++;
0
1        var goods_id=_h_goods_id[array_index].value;
2        var fileName=_h_goods_fileName[array_index].value;
3        img_sticky.src=SERVER_URL+"?goods_id="+goods_id+"&fileName="+fileName;
4        cur_goods_num.innerHTML=array_index+1;
5    }
6
7
8    function fn_show_previous_goods(){
9        var img_sticky=document.getElementById("img_sticky");
0        var cur_goods_num=document.getElementById("cur_goods_num");
1        var _h_goods_id=document.frm_sticky.h_goods_id;
2        var _h_goods_fileName=document.frm_sticky.h_goods_fileName;
3
4        if(array_index >0)
5            array_index--;
6
7        var goods_id=_h_goods_id[array_index].value;
8        var fileName=_h_goods_fileName[array_index].value;
9        img_sticky.src=SERVER_URL+"?goods_id="+goods_id+"&fileName="+fileName;
0        cur_goods_num.innerHTML=array_index+1;
1    }

```



```

function goodsDetail(){
    var cur_goods_num=document.getElementById("cur_goods_num");
    arrIdx=cur_goods_num.innerHTML-1;

    var img_sticky=document.getElementById("img_sticky");
    var h_goods_id=document.frm_sticky.h_goods_id;
    var len=h_goods_id.length;

    if(len>1){
        goods_id=h_goods_id[arrIdx].value;
    }else{
        goods_id=h_goods_id.value;
    }

    var formObj=document.createElement("form");
    var i_goods_id = document.createElement("input");

    i_goods_id.name="goods_id";
    i_goods_id.value=goods_id;

    formObj.appendChild(i_goods_id);
    document.body.appendChild(formObj);
    formObj.method="get";
    formObj.action="${contextPath}/goods/goodsDetail.do?goods_id="+goods_id;
    formObj.submit();
}

</script>

```

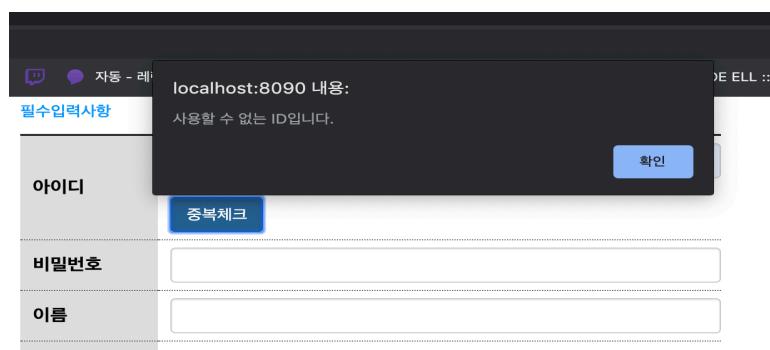
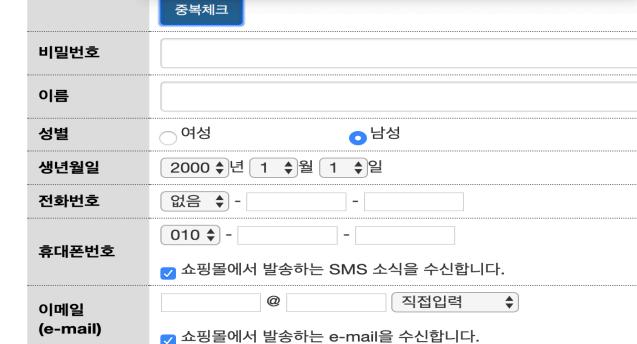
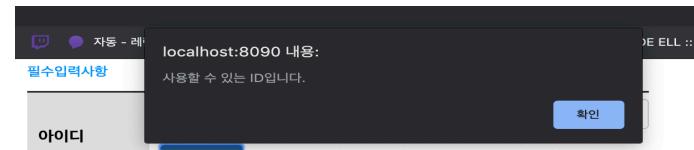
이미지 클릭 시 해당 상품  
상세 화면으로 이동



## 회원가입 화면

**필수입력사항**

아이디	ajh3166	<b>중복체크</b>
비밀번호		
이름		
성별	<input type="radio"/> 여성	<input checked="" type="radio"/> 남성
생년월일	2000	년 1 월 1 일
전화번호	없음 - - -	
휴대폰번호	010 - - -	
<input checked="" type="checkbox"/> 쇼핑몰에서 발송하는 SMS 소식을 수신합니다. <input checked="" type="checkbox"/> 쇼핑몰에서 발송하는 e-mail을 수신합니다.		
이메일 (e-mail)	@	직접입력
<input type="text"/> @ <input type="text"/> 직접입력		
<a href="#">우편번호검색</a> 지번 주소: <input type="text"/>		
주소	도로명 주소: <input type="text"/>  나머지 주소: <input type="text"/>	
<a href="#">회원 가입</a> <a href="#">다시입력</a>		



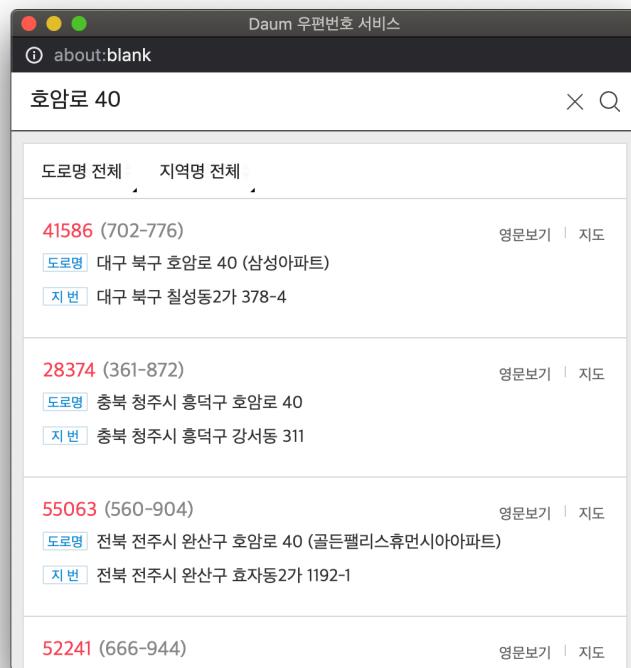
중복 계정 체크

# memberForm.jsp(2)

중복 체크 클릭 시 비동기.ajax로 데이터 전송)

```
function fn_overlapped(){
    var _id=$("#_member_id").val();
    if(_id==''){
        alert("ID를 입력하세요");
        return;
    }
    $.ajax({
        type:"post",
        async:false,
        url:"${contextPath}/member/overlapped.do",
        dataType:"text",
        data: {id:_id},
        success:function (data,textStatus){
            if(data=='false'){
                alert("사용할 수 있는 ID입니다.");
                /* $('#btnOverlapped').prop("disabled", true);
                $('#_member_id').prop("disabled", true); */
                $('#member_id').val(_id);
            }else{
                alert("사용할 수 없는 ID입니다.");
            }
        },
        error:function(data,textStatus){
            alert("에러가 발생했습니다.");
        },
        complete:function(data,textStatus){
            //alert("작업을완료 했습니다");
        }
    });
} //end ajax
```

주소 검색 Daum api 사용



주소 클릭 시 입력

## 주소 검색 Daum api 사용

```

function execDaumPostcode() {
    new daum.Postcode({
        oncomplete: function(data) {
            // 팝업에서 검색결과 항목을 클릭했을때 실행할 코드를 작성하는 부분.
            // 도로명 주소의 노출 규칙에 따라 주소를 조합한다.
            // 내려오는 변수가 값이 없는 경우엔 공백('')값을 가지므로, 이를 참고하여 분기 한다.
            var fullRoadAddr = data.roadAddress; // 도로명 주소 변수
            var extraRoadAddr = ''; // 도로명 조합형 주소 변수

            // 법정동명이 있을 경우 추가한다. (법정리는 제외)
            // 법정동의 경우 마지막 문자가 "동|로|가"로 끝난다.
            if(data.bname !== '' && /[동|로|가]/g.test(data.bname)){
                extraRoadAddr += data.bname;
            }
            // 건물명이 있고, 공동주택일 경우 추가한다.
            if(data.buildingName !== '' && data.apartment === 'Y'){
                extraRoadAddr += (extraRoadAddr !== '' ? ', ' + data.buildingName : data.buildingName);
            }
            // 도로명, 지번 조합형 주소가 있을 경우, 괄호까지 추가한 최종 문자열을 만든다.
            if(extraRoadAddr !== ''){
                extraRoadAddr = ' (' + extraRoadAddr + ')';
            }
            // 도로명, 지번 주소의 유무에 따라 해당 조합형 주소를 추가한다.
            if(fullRoadAddr !== ''){
                fullRoadAddr += extraRoadAddr;
            }

            // 우편번호와 주소 정보를 해당 필드에 넣는다.
            document.getElementById('zipcode').value = data.zonecode; // 5자리 새우편번호 사용
            document.getElementById('road_address').value = fullRoadAddr;
            document.getElementById('jibun_address').value = data.jibunAddress;
            // 사용자가 '선택 안함'을 클릭한 경우, 예상 주소라는 표시를 해준다.
            if(data.autoRoadAddress) {
                //예상되는 도로명 주소에 조합형 주소를 추가한다.
                var expRoadAddr = data.autoRoadAddress + extraRoadAddr;
                document.getElementById('guide').innerHTML = '(예상 도로명 주소 : ' + expRoadAddr + ')';
            } else if(data.autoJibunAddress) {
                var expJibunAddr = data.autoJibunAddress;
                document.getElementById('guide').innerHTML = '(예상 지번 주소 : ' + expJibunAddr + ')';
            } else {
                document.getElementById('guide').innerHTML = '';
            }
        }
    }).open();
}

```

<tr class="dot_line"> <td class="fixed_join">주소</td> <td> <input class="form-control" id="zipcode" name="zipcode" size="10" style="width: 100px;" type="text"/> <a href="javascript:execDaumPostcode()">우편번호검색</a>   <p>지번 주소:&lt;br&gt;&lt;input class="form-control" type="text" id="road_address" name="road_address" size="50"&gt;&lt;br&gt;도로명 주소: &lt;input class="form-control" type="text" id="jibun_address" name="jibun_address" size="50"&gt;&lt;br&gt;나머지 주소: &lt;input class="form-control" type="text" name="detail_address" size="50" /&gt;</p> <span id="guide" style="color:#999;">(예상 주소)</span> </td> </tr>	주소	<input class="form-control" id="zipcode" name="zipcode" size="10" style="width: 100px;" type="text"/> <a href="javascript:execDaumPostcode()">우편번호검색</a> <p>지번 주소:&lt;br&gt;&lt;input class="form-control" type="text" id="road_address" name="road_address" size="50"&gt;&lt;br&gt;도로명 주소: &lt;input class="form-control" type="text" id="jibun_address" name="jibun_address" size="50"&gt;&lt;br&gt;나머지 주소: &lt;input class="form-control" type="text" name="detail_address" size="50" /&gt;</p> <span id="guide" style="color:#999;">(예상 주소)</span>
주소	<input class="form-control" id="zipcode" name="zipcode" size="10" style="width: 100px;" type="text"/> <a href="javascript:execDaumPostcode()">우편번호검색</a> <p>지번 주소:&lt;br&gt;&lt;input class="form-control" type="text" id="road_address" name="road_address" size="50"&gt;&lt;br&gt;도로명 주소: &lt;input class="form-control" type="text" id="jibun_address" name="jibun_address" size="50"&gt;&lt;br&gt;나머지 주소: &lt;input class="form-control" type="text" name="detail_address" size="50" /&gt;</p> <span id="guide" style="color:#999;">(예상 주소)</span>	

## 메인 페이지 화면

## 스포츠룩 / 데일리룩 완성!

아디다스 나이키 외 인기브랜드 매일 업데이트



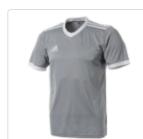
인기상품

팀 플립 19 폴리스 크루 L/S(기모)(AJ1466-010)(146601)  
50,000원예센셜 스웨트 크루 18 L/S(DN1451)  
55,000원파크 2 니트 소트 WB(725903-657)  
(725903657)  
30,000원맨유 17-18 홈 소트(BQ3739)  
30,000원아디다스 프레데터19.2 FG 축구화  
(BB8111)  
200,000원레볼라 2 V1 JAPAN(P1GA197001)  
240,000원나이키 티эм포 레전드7 엘리트 FG  
200,000원팀 플립 19 폴리스 크루 L/S(기모)(AJ1466-010)  
60,000원

## 스티디셀러

레알 마드리드 미디엄 더플(2방)(DY7713)  
50,000원플레이어 우븐 탑(RED)  
50,000원미즈노 하프 칭업 니트 자켓 19(KO209)  
40,000원세리노 19 트레이닝 탑 L/S(DY3145)  
50,000원세리노 19 트레이닝 탑 L/S(DY3146)  
50,000원AS 나이키 써미 아카데미 클럽 탑 L/S(기모)(AO9190-011)(9190011)  
40,000원나이키 드라이 아카데미19 KZ 소트  
(AJ9126-461)  
40,000원아디다스 엑스 클립 신기드(DY0088)  
12,000원로이어 피트컨트롤 프로G3-피드백프 골키  
피장갑(S970979704)  
80,000원아디다스 아디다스X 씨커스타킹(CF3581)  
12,000원아디다스 프레데터 19.4 FKG 축구화  
(F35597)  
250,000원프레데터 19.1 HG 축구화(EF8994)  
200,000원

## 새로 출시한 제품

맨유 18-19 홈 소트(CG0042)  
35,000원유벤투스 18-19 홈 저지 S/S(CF3489)  
200,000원맨유 18-19 홈 저지 S/S(CG0040)  
220,000원바르셀로나 19-20 BRT 스티디움 홈 저지  
S/S(AJ5532-456)  
200,000원아디다스 타렐라18 반팔저지(CE8940)  
32,000원아디다스 코파19.1 HG 축구화(EF9019)  
250,000원푸마 원 18.1 FG 축구화(10452706)  
200,000원프레데터 19.1 HG(EF8994)  
220,000원

## 메인 페이지 jsp

```

<div class="main_book">
    <:set var="goods_count" value="0" />
    <h3>인기상품</h3>
    <c:forEach var="item" items="${goodsMap.bestseller }">
        <c:set var="goods_count" value="${goods_count+1 }" />
        <div class="book">
            <a href="${contextPath}/goods/goodsDetail.do?goods_id=${item.goods_id }">
                
            </a>
            

            <div class="title">${item.goods_title }</div>
            <div class="price">
                <fmt:formatNumber value="${item.goods_price}" type="number" var="goods_price" />
                ${goods_price}원
            </div>
        </div>
        <c:if test="${goods_count==15 }">
            <div class="book">
                <font size=20> <a href="#">more</a></font>
            </div>
        </c:if>
    </c:forEach>
</div>
<div class="clear"></div>

```

상품 status 조건에 따라 상품을 보여줌

```

        <div id="ad_sub_banner">
            
        </div>
        <div class="main_book" >
            <c:set var="goods_count" value="0" />
            <h3>새로 출시한 제품</h3>
            <c:forEach var="item" items="${goodsMap.newproduct }" >
                <c:set var="goods_count" value="${goods_count+1 }" />
                <div class="book">
                    <a href="${contextPath}/goods/goodsDetail.do?goods_id=${item.goods_id }">
                        
                    </a>
                    
                    <div class="title">${item.goods_title }</div>
                    <div class="price">
                        <fmt:formatNumber value="${item.goods_price}" type="number" var="goods_price" />
                        ${goods_price}원
                    </div>
                </div>
                <c:if test="${goods_count==15 }">
                    <div class="book">
                        <font size=20> <a href="#">more</a></font>
                    </div>
                </c:if>
            </c:forEach>
        </div>

```

상품 상세 페이지 화면

FootBall Mall 로그인 회원가입  검색

**축구용품**

- 축구화
- 유니폼 상의
- 유니폼 하의
- 크루탑
- 트레이닝복
- 악세사리



**축구용품**

**에센셜 스웨트 크루 18 L/S(DN1451)**

아디다스



**정가** 55,000원

**판매가** 49,500원(10% 할인)

**포인트적립** 550P(10% 적립)

**포인트 추가적립** 만원이상 구매시 1,000P, 5만원이상 구매시 2,000P 추가적립 편의점 배송 이용시 300P 추가적립

**발행일** 2019-02-27

**배송료** 무료

**배송안내** [당일배송] 당일배송 서비스 시작!  
[휴일배송] 휴일에도 배송받는 쇼핑몰

**도착예정일** 지금 주문 시 내일 도착 예정

**수량**

**사이즈**

구매하기
장바구니
위시리스트

상품 소개
교환/반품/배송



최근 본 상품



이전 1/1

## 상품 상세 페이지 화면

상품 소개 교환/반품/배송



## 상세 이미지



상품 소개 교환/반품/배송

## 비밀번호

- 배송기간은 평균 결제 완료 후 1~3일 정도 소요되며, 제품에 따라 약간의 차이가 생길 수 있습니다.
- 배송비는 전국 2,500원의 배송비가 부과됩니다.(제주도,도서 지방은 별도 요금 부과)
- 배송 업체는 "CJ대한통운"이며, 평일 오후 5시까지 발송 업무를 처리 합니다.토요일,일요일,공휴일은 발송처리가 되지 않습니다.

## 비밀번호 찾기 및 초기화

- 제품페이지에 의한 교환/반품 배송받는 쇼핑몰에서 전역 부담합니다.
  - 제품페이지 아닌 교환/반품의 경우 배송받는 전역 고객님께서 부담하셔야 합니다.
- (교환시 흥제배송비: 6,000원, 반품시 편도배송비: 3,000원)

## 교환/반품

- 교환/반품 요청서를 등봉하고 제품, 사용품 및 정품박스가 손상되지 않도록 잘 포장한뒤 반드시 CJ대한통운 택배점수(☎ 1588-1255, 편의점 택배 제외)사 방문시 해당제품을 보내주시면 됩니다.
- 교환시 흥제배송비 6,000원 / 반품시 편도배송비 3,000원을 입금하시면 교환,반품이 진행됩니다.(제주도,도서 지방은 추가 요금 발생)
- 티백에 이동시 추가비용이 발생합니다.
- 편도 배송비를 초과하는 일부 대형제품의 경우 배송비가 추가됩니다.

## 반품 및 교환이 불가능한 경우

- 제품을 사용하거나 훼손된 흔적이 있는 경우
- 상품 수령일로부터 7일이 경과한 경우
- 제품에 붙어있는 'tag'를 떼어 내거나 정품포장박스, 비닐이 훼손된 경우
- 미단을 추가로 하신 제품의 경우
- 주문 생산 제품 및 제품 상세설명에 교환/반품 불가능이라 명시된 제품의 경우

## 정품박스 훼손시 교환/반품불가!!!

박스 훼손 반품시 임의반송 처리 될 수 있습니다.

교환,환불시 정물박스를 꼭 이중포장 해주세요.(처음 받아보셨던 상태와 동일하게 포장)

사이즈 교환은 많은 신발과 의류의 특성상 저희로서도 정말 어려운 부분입니다.

나이키 코리아,아디다스 코리아에서 정품으로 인증을 받을 수 있는 방법은 텍(라벨)과 정품박스 혹은 포장비닐에 붙어있는 품질표시 정보들을 병법밖에 없습니다.

# goodsDetail.jsp(코드)

상품 상세 페이지 화면

```

</figure>
</div>
<div id="detail_table">
<table>
<tbody>
<tr>
    <td class="fixed">정가</td>
    <td class="active"><span >
        <fmt:formatNumber value="${goods.goods_price}" type="number" var="goods_price" />
        ${goods_price}원
    </span></td>
</tr>
<tr class="dot_line">
    <td class="fixed">판매가</td>
    <td class="active"><span >
        <fmt:formatNumber value="${goods.goods_price*0.9}" type="number" var="discounted_price" />
        ${discounted_price}원(10%할인)</span></td>
</tr>
<tr>
    <td class="fixed">포인트적립</td>
    <td class="active">${goods.goods_point}P(10%적립)</td>
</tr>
<tr class="dot_line">
    <td class="fixed">포인트 추가적립</td>
    <td class="fixed">만원이상 구매시 1,000P, 5만원이상 구매시 2,000P추가적립 편의점 배송 이용시 300P 추가적립</td>
</tr>
<tr>
    <td class="fixed">별행일</td>
    <td class="fixed">
        <c:set var="pub_date" value="${goods.goods_production_date}" />
        <c:set var="arr" value="${fn:split(pub_date, ' ')}" />
        <c:out value="${arr[0]}" />
    </td>
</tr>
```

썸네일 이미지 : FileDownLoadController.java의 controller 호출

## 장바구니 담기 클릭

```

function add_cart(goods_id) {
    var order_goods_qty = document.getElementById("order_goods_qty").value;
    var order_goods_size = document.getElementById("order_goods_size").value;
    console.log(order_goods_qty);
    console.log(order_goods_size);
    $ajax({
        type : "post",
        async : false, //false인 경우 동기식으로 처리한다.
        url : "${contextPath}/cart/addGoodsInCart.do",
        data : {
            goods_id:goods_id,
            order_goods_qty: order_goods_qty,
            order_goods_size:order_goods_size
        },
        success : function(data, textStatus) {
            if(data.trim()=='add_success'){
                imagePopup('open', '.layer01');
            }else if(data.trim()=='already_existed'){
                alert("이미 카트에 등록된 상품입니다.");
            }
        },
        error : function(data, textStatus) {
            alert("에러가 발생했습니다." +data);
        },
        complete : function(data, textStatus) {
        }
    }); //end ajax
}

```

비동기로 상품 정보 controller에 데 이터 전송

```

function imagePopup(type) {
    if (type == 'open') {
        // 팝업창을 연다.
        jQuery('#layer').attr('style', 'visibility:visible');

        // 페이지를 가리기위한 레이어 영역의 높이를 페이지 전체의 높이와 같게 한다.
        jQuery('#layer').height(jQuery(document).height());
    }

    else if (type == 'close') {

        // 팝업창을 닫는다.
        jQuery('#layer').attr('style', 'visibility:hidden');
    }
}

```

## 축+봉 씀

파크 2 니트 쇼트 WB(725903-657)(725903657)

나이키



정가	30,000원
판매가	27,000원(10% 할인)
포인트적립	300P(10% 적립)
포인트 추가적립	만원이상 구매시 1,000P, 5만원이상 구매시 2,000P 추가적립 편의점 배송 이용시 300P 추가적립
발행일	2019-04-04
배송료	무료
배송안내	[당일내][후일내]
도착예정일	지금 바로
수량	1
사이즈	L

구매하기
장바구니

장바구니에 담았습니다.
  
장바구니 보기

구매하기 클릭

```
function fn_order_each_goods(goods_id,goods_title,goods_sales_price,fileName){
    var _isLogOn=document.getElementById("isLogOn");
    var isLogOn=_isLogOn.value;

    if(isLogOn=="false" || isLogOn==''){
        alert("로그인 후 주문이 가능합니다!!!!");
    }

    var total_price,final_total_price;
    var order_goods_qty=document.getElementById("order_goods_qty");
    var order_goods_size=document.getElementById("order_goods_size");

    var formObj=document.createElement("form");
    var i_goods_id = document.createElement("input");
    var i_goods_title = document.createElement("input");
    var i_goods_sales_price=document.createElement("input");
    var i_fileName=document.createElement("input");
    var i_order_goods_qty=document.createElement("input");
    var i_order_goods_size=document.createElement("input");

    i_goods_id.name="goods_id";
    i_goods_title.name="goods_title";
    i_goods_sales_price.name="goods_sales_price";
    i_fileName.name="goods_fileName";
    i_order_goods_qty.name="order_goods_qty";
    i_order_goods_size.name="order_goods_size";

    i_goods_id.value=goods_id;
    i_order_goods_qty.value=order_goods_qty.value;
    i_order_goods_size.value=order_goods_size.value;
    i_goods_title.value=goods_title;
    i_goods_sales_price.value=goods_sales_price;
    i_fileName.value=fileName;
```

```
formObj.appendChild(i_goods_id);
formObj.appendChild(i_goods_title);
formObj.appendChild(i_goods_sales_price);
formObj.appendChild(i_fileName);
formObj.appendChild(i_order_goods_qty);
formObj.appendChild(i_order_goods_size);

document.body.appendChild(formObj);
formObj.method="post";
formObj.action="${contextPath}/order/orderEachGoods.do";
formObj.submit();
```

button의 onclick 속성을 사용하여 클릭 시 form을 생성  
하여 post 방식으로 orderEachGoods.do 호출

02

본론

# orderGoodsForm.jsp

구매하기 버튼 클릭 시 주문 화면으로 이동

localhost:8090/biz/order/orderEachGoods.do

1. 주문 확인

축구용품	주문상품명	수량	사이즈	주문금액	배송비	예상적립금	주문금액합계
	파크 2 니트 소트 WB(725903-657) [725903657]	1개	L	30000원 (10% 할인)	0원	1500원	30000원



2. 배송지 정보

배송방법	<input checked="" type="radio"/> 일반택배 <input type="radio"/> 편의점택배 <input type="radio"/> 해외배송
배송지 선택	<input checked="" type="radio"/> 기본배송지 <input type="radio"/> 새로입력 <input type="radio"/> 최근배송지
받으실 분	안지혁
휴대폰번호	010 - 1111 - 11111
유선전화(선택)	02 - 222 - 2222
주소	13536 우편번호검색 도로명 주소: 경기 성남시 분당구 판교역로 4 (백현동)
배송 메시지	택배 기사님께 전달할 메시지를 남겨주세요.
선물 포장	<input type="radio"/> 예 <input checked="" type="radio"/> 아니요

축구용품계의 신 뜻불을!!




로그 아웃 상태에서 클릭 시

localhost:8090/biz/goods/goodsDetail.do?goods\_id=586

localhost:8090 내용:  
로그인 후 주문이 가능합니다!!!

FootBall Mall

확인

축구용품

예상 금액: 55,000원

주문내용:

축구용품	예상 금액: 55,000원
주문상품명	판매가: 49,500원 (10% 할인)
수량	포인트 추가적립: 550P (10% 적립)
사이즈	포인트 추가적립: 550P (10% 적립)
발행일	만일배송 구매시 1,000P, 5만원이상 구매시 2,000P 추가적립 편의점 배송 이용시 300P 추가적립
배송료	만일배송 서비스 시작!
배송안내	[당일배송] 당일배송 서비스 시작! [숙박배송] 휴일에도 배송받는 소원을
도착예정일	지금 주문 시 내일 도착 예정
수량	구매하기
사이즈	장바구니

상품 소개

교환/반품/배송

최근 본 상품

이전 1/1 다음

축구용품계의 신 뜻불을!!





02

본론

# orderGoodsForm.jsp

alert 창 확인 클릭 후 로그인 화면으로 이동

바로 주문 화면으로 이동

회원 로그인 창

아이디 korea01
비밀번호 ***

Remember me

로그인

회원가입 메인

마이페이지 장바구니 로그아웃 검색

1. 주문 확인

주문상품명	수량	사이즈	주문금액	배송비	예상적립금	주문금액합계
	1개	L	55000원 (10% 할인)	0원	1500원	55000원

2. 배송지 정보

배송방법	<input checked="" type="radio"/> 일반택배 <input type="radio"/> 편의점택배 <input type="radio"/> 해외배송	주문고객
배송지 선택	<input checked="" type="radio"/> 기본배송지 <input type="radio"/> 새로입력 <input type="radio"/> 최근배송지	
받으실 분	안지혁	이름
휴대폰번호	010 <input type="button" value="▼"/> - <input type="text" value="1111"/> - <input type="text" value="11111"/>	핸드폰
유선전화(선택)	02 <input type="button" value="▼"/> - <input type="text" value="222"/> - <input type="text" value="2222"/>	이메일
13536   우편번호검색		

02

## 본론

## 장바구니(myCartList.jsp)

여러 상품 장바구니에 담은 화면

구분	상품명	정가	판매가	사이즈	수량	합계	주문
<input checked="" type="checkbox"/>	파크 2 라이트 쇼트 WB(725903-657)(725903657)	30000원	27,000원(10% 할인)	L	<input type="text" value="1"/> 변경	27,000원	<button>주문하기</button>
<input checked="" type="checkbox"/>	맨유 17-18 홈 쇼트(BQ3739)	30000원	27,000원(10% 할인)	L	<input type="text" value="1"/> 변경	27,000원	<button>주문하기</button>

주문하기 쇼핑계속하기

장바구니 담은 상품 수량 변경

localhost:8090 내용:  
수량을 변경했습니다!!

구분	상품명	정가	판매가	사이즈	수량	합계	주문
<input checked="" type="checkbox"/>	파크 2 라이트 쇼트 WB(725903-657)(725903657)	30000원	27,000원(10% 할인)	L	<input type="text" value="2"/> 변경	27,000원	<button>주문하기</button>
<input checked="" type="checkbox"/>	맨유 17-18 홈 쇼트(BQ3739)	30000원	27,000원(10% 할인)	L	<input type="text" value="1"/> 변경	27,000원	<button>주문하기</button>

총 상품 수 : 상품 종류 별 수

주문하기 쇼핑계속하기

최종 결제 금액 변경

# 상품주문(orderGoodsForm.jsp)

2개 이상 주문 시 orderAllCartGoods.do 호출  
(상품 상세페이지에서 주문바로하기 화면과 같으나 url주소는 다름)

① localhost:8090/biz/order/orderAllCartGoods.do

## 1. 주문 확인

주문상품명	수량	사이즈	주문금액	배송비	예상적립금	주문금액합계
	2개	L	30000원 (10% 할인)	0원	3000원	60000원
	1개	L	30000원 (10% 할인)	0원	1500원	30000원

## 2. 배송지 정보

배송방법	<input checked="" type="radio"/> 일반택배 <input type="radio"/> 편의점택배 <input type="radio"/> 해외배송
배송지 선택	<input checked="" type="radio"/> 기본배송지 <input type="radio"/> 새로입력 <input type="radio"/> 최근배송지
받으실 분	안지혁
휴대폰번호	010-1111-1111
유선전화(선택)	02-222-2222
13536   우편번호검색 도로명 주소: 경기 성남시 분당구 판교역로 4 (백현동)	
주소	지번 주소: 경기 성남시 분당구 백현동 582-7
나머지 주소: 우방아파트 101동 1604호	
배송 메시지	택배 기사님께 전달할 메시지를 남겨주세요.

## 3. 할인 정보

적립금	원/1000원	<input type="checkbox"/> 모두 사용하기
예치금	원/1000원	<input type="checkbox"/> 모두 사용하기
상품권 전환금	원/0원	<input type="checkbox"/> 모두 사용하기
OK 캐쉬백 포인트	원/0원	<input type="checkbox"/> 모두 사용하기
쿠폰할인	원/0원	<input type="checkbox"/> 모두 사용하기

총 상품수	총 상품금액	총 배송비	총 할인 금액	최종 결제금액
3개	90000원	+ 0원	- 0원	= 90000원

## 4. 결제 정보

<input checked="" type="radio"/> 신용카드	<input type="radio"/> 카카오카드	<input type="radio"/> 실시간 계좌이체	<input type="radio"/> 무통장 입금
<input type="radio"/> 휴대폰 결제	<input type="radio"/> 키카오페이(간편결제)	<input type="radio"/> 페이나우(간편결제)	<input type="radio"/> 페이코(간편결제)
<input type="radio"/> 직접입금			
카드 선택:	삼성		
할부 기간:	일시불		

결제하기

쇼핑계속하기

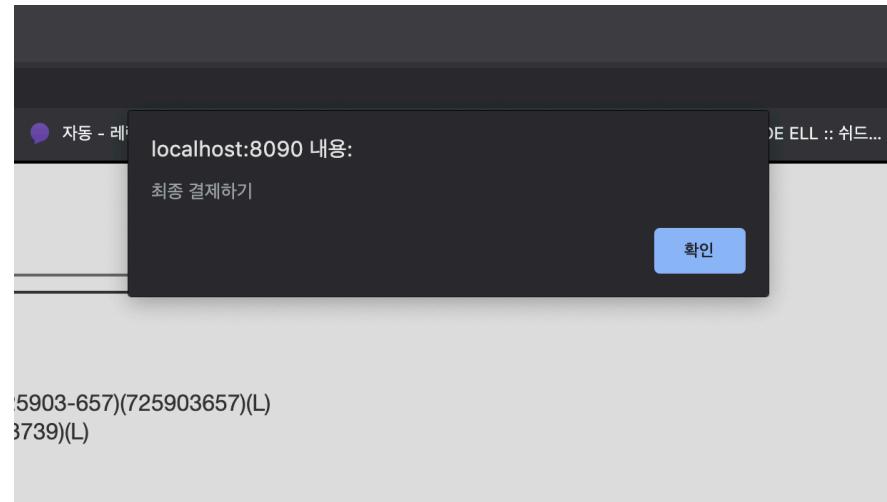
결제 하기 클릭 시 팝업 화면

최종 주문 사항

주문상품번호:	585 584
주문상품명:	파크 2 니트 소트 WB(725903-657)(725903657)(L) 멘유 17-18 흠 쇼트(BQ3739)(L)
주문상품개수:	3개
주문금액합계:	90000원
주문자:	안지혁
받는사람:	안지혁
배송방법:	일반택배
받는사람 휴대폰번호:	010-1111-1111
받는사람 유선번화번호:	02-222-2222
우편번호:	13536
배송주소:	도로명 주소: 경기 성남시 분당구 판교역로 4 (백현동) [지번 주소: 경기 성남시 분당구 백현동 582-7] 우방아파트 101동 1604호
배송메시지:	
선물포장 여부:	no
결제방법:	신용카드 카드사:삼성 할부개월수:일시불

[최종결제하기](#)

최종결제하기 클릭 시 화면



## 주문 결과화면

## 1. 최종 주문 내역서

주문번호	주문상품명	수량	사이즈	주문금액	배송비	예상적립금	주문금액합계
449	파크 2 니트 소트 WB(725903-657)(725903657)	L	2개	60000원 (10% 할인)	0원	3000원	60000원
449	멘유 17-18 홈 소트(BQ3739)	L	1개	30000원 (10% 할인)	0원	1500원	30000원

## 2. 배송지 정보

배송방법	일반택배
받으실 분	안지혁
휴대폰번호	010-1111-1111
유선전화(선택)	02-222-2222
주소	우편번호:13536 도로명 주소:경기 성남시 분당구 판교역로 4 (백현동) [지번 주소:경기 성남시 분당구 백현동 582-7] 우방아파트 101동 1604호
배송 메시지	
선물 포장	no

## 3. 결제 정보

결제방법	신용카드 카드사:삼성 할부개월수:일시불
결제카드	삼성
할부기간	일시불

쇼핑계속하기

## 관리자 페이지 메인화면

**상품 조회**

간단조회  상세조회

2019년 6월 19일 이전

당일 1주 2주 1개월 2개월 3개월 4개월 까지 조회

**✓ 상품번호**

상품이름 제조

2019년 05월 12일 ~ 2019년 06월 19일

상품번호	상품이름	제조사	상품가격	상품 색상	상품 사이	상품 수량	입고일자	제조일
594	레알 마드리드 미디엄 더플(기방)(DY7713)	아디다스	50000	black	M	20	2019-06-18	2019-05-08
592	플레이어 우븐 탑(RED)	아스토레	50000	red	L	20	2019-06-18	2019-05-02
592	플레이어 우븐 탑(RED)	아스토레	50000	red	XL	20	2019-06-18	2019-05-02
591	미즈노 하프 짚업 니트 자켓 1(9K0209)	미즈노	40000	black	M	20	2019-06-18	2019-01-29
591	미즈노 하프 짚업 니트 자켓 1(9K0209)	미즈노	40000	black	S	20	2019-06-18	2019-01-29
591	미즈노 하프 짚업 니트 자켓 1(9K0209)	미즈노	40000	black	L	20	2019-06-18	2019-01-29
590	세레노 19 트레이닝 탑 L/S(DY3145)	아디다스	50000	navy	M	20	2019-06-18	2019-05-28
590	세레노 19 트레이닝 탑 L/S(DY3145)	아디다스	50000	navy	L	20	2019-06-18	2019-05-28
590	세레노 19 트레이닝 탑 L/S(DY3145)	아디다스	50000	navy	S	20	2019-06-18	2019-05-28
589	세레노 19 트레이닝 탑 L/S(DY3146)	아디다스	50000	navy	L	20	2019-06-18	2019-05-29

1 2 3 4 5 6 7 8 9 10 next

**상품 등록하기**

# 관리자(adminGoodsMain.jsp)

## 관리자 페이지 간단조회

localhost:8090/biz/admin/goods/adminGoodsMain.do?beginDate=2019-06-19&endDate=2019-06-19

FootBall Mall

マイ페이지 장바구니 관리자 로그아웃 검색

**상품 조회**

간단조회  상세조회

2019 년 6 월 19 일 이전

당일 1주 2주 1개월 2개월 3개월 4개월 까지 조회

상품번호 조회

조회 기간: 2019 년 06 월 19 일 ~ 2019 년 06 월 19 일

상품번호 상품이름 제조사 상품가격 상품 색상 상품 사이 상품 수량 입고일자 제조일

최근 본 상품  
상품이 없습니다.

0/0

조회된 상품이 없습니다.

1 2 3 4 5 6 7 8 9 10 next

상품 등록하기



# 관리자(adminGoodsMain.jsp)(1)

관리자 페이지 간단 조회 1주 클릭 시

localhost:8090/biz/admin/goods/adminGoodsMain.do?beginDate=2019-05-05&endDate=2019-06-19

FootBall Mall 마이페이지 장바구니 관리자 로그아웃 검색

**상품 조회**

간단조회 상세조회  
 2019 년 6 월 19 일 이전      당일 1주 2주 1개월 2개월 3개월 4개월      까지 조회  
 상품번호      조회  
 조회 기간: 2019 년 05 월 05 일 ~ 2019 년 06 월 19 일

상품번호	상품이름	제조사	상품가격	상품 색상	상품 사이	상품 수량	입고일자	제조일
594	레알 마드리드 미디엄 더플(기방)(DY7713)	아디다스	50000	black	M	20	2019-06-18	2019-05-08
592	플레이어 우븐 탑(RED)	아스토레	50000	red	L	20	2019-06-18	2019-05-02
592	플레이어 우븐 탑(RED)	아스토레	50000	red	XL	20	2019-06-18	2019-05-02
591	미즈노 하프 짚업 니트 자켓 1(9K0209)	미즈노	40000	black	M	20	2019-06-18	2019-01-29
591	미즈노 하프 짚업 니트 자켓 1(9K0209)	미즈노	40000	black	S	20	2019-06-18	2019-01-29
591	미즈노 하프 짚업 니트 자켓 1(9K0209)	미즈노	40000	black	L	20	2019-06-18	2019-01-29
590	세레노 19 트레이닝 탑 L/S(DY3145)	아디다스	50000	navy	M	20	2019-06-18	2019-05-28
590	세레노 19 트레이닝 탑 L/S(DY3145)	아디다스	50000	navy	L	20	2019-06-18	2019-05-28
590	세레노 19 트레이닝 탑 L/S(DY3145)	아디다스	50000	navy	S	20	2019-06-18	2019-05-28
589	세레노 19 트레이닝 탑 L/S(DY3146)	아디다스	50000	navy	L	20	2019-06-18	2019-05-29

1 2 3 4 5 6 7 8 9 10 next

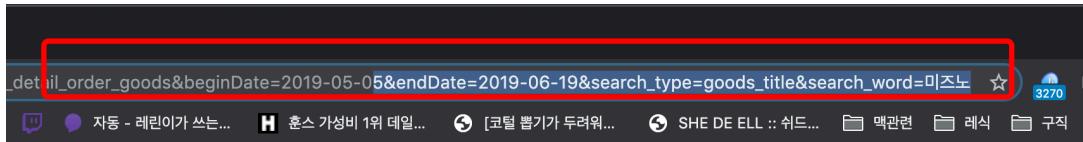
축구용품계의 신 풋볼!!

상품 등록하기



# 관리자(adminGoodsMain.jsp)(2)

## 관리자 페이지 상세 조회



**상품 조회**

간단조회  상세조회

2019  년 6  월 19  일 이전        까지 조회

상품번호

조회 기간: 2019  년 05  월 05  일 ~ 2019  년 06  월 19

상품번호	상품이름	제조사	상품가격	상품 색상	상품 사이	상품 수량	입고일자	제조일
591	미즈노 하프 짧업 니트 자켓 1(9K0209)	미즈노	40000	black	L	20	2019-06-18	2019-01-29
591	미즈노 하프 짧업 니트 자켓 1(9K0209)	미즈노	40000	black	S	20	2019-06-18	2019-01-29
591	미즈노 하프 짧업 니트 자켓 1(9K0209)	미즈노	40000	black	M	20	2019-06-18	2019-01-29

1 2 3 4 5 6 7 8 9 10 next

[상품 등록하기](#)

페이스북  
인스타 그램  
트위터

최근 본 상품  
상품이 없습니다

0/0

## 관리자 상품 등록 페이지

새상품 등록창

상품정보      상품소개      상품이미지

제품분류	✓ 축구화 크루탐 유니폼 상의 유니폼 하의 트레이닝복 악세사리
제품이름	
제조사	
제품정가	
제품판매가격	
제품 구매 포인트	
제품제조 날짜	연도. 월. 일.
제품 배송비	
제품 도착 예정일	연도. 월. 일.
제품종류	신제품
제품 색상	
재고 수량	
사이즈	<input type="checkbox"/> 250 <input type="checkbox"/> 260 <input type="checkbox"/> 270 <input type="checkbox"/> 280 <input type="checkbox"/> S <input type="checkbox"/> M <input type="checkbox"/> L <input type="checkbox"/> XL

**상품 등록하기**

새상품 등록창

상품정보      상품소개      상품이미지

이미지파일 첨부	파일 선택      나이키 드라이 ...AJ9128-451).jpg
	파일 선택      나이키 드라이 아...8-451)_상세.jpg

**상품 등록하기**

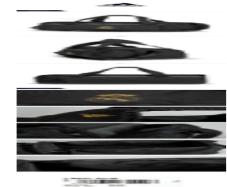
<https://github.com/zidol/SpringFootballMall/blob/master/src/main/webapp/WEB-INF/views/admin/goods/addNewGoodsForm.jsp>

상품 이름 클릭 시 수정 화면으로 이동

	상품정보	상품소개	상품이미지
상품분류	악세사리 ▼	<input type="button" value="수정"/>	
상품이름	레알 마드리드 미디엄 더플(가방)(DY7713)	<input type="button" value="수정"/>	
제조사	아디다스	<input type="button" value="수정"/>	
제품 사이즈	M	<input type="button" value="수정"/>	
제품 색상	black	<input type="button" value="수정"/>	
재고 수량	20	<input type="button" value="수정"/>	
상품정가	50000	<input type="button" value="수정"/>	
상품판매가격	50000	<input type="button" value="수정"/>	
상품 구매 포인트	500	<input type="button" value="수정"/>	
상품출판일	2019. 05. 08.	<input type="button" value="수정"/>	
상품 배송비	3000	<input type="button" value="수정"/>	
상품 도착 예정일	2019. 06. 11.	<input type="button" value="수정"/>	
상품종류	스테디셀러 ▼	<input type="button" value="수정"/>	

상품정보      상품소개      상품이미지

메인 이미지  선택된 파일 없음 

상세 이미지1  선택된 파일 없음 

<https://github.com/zidol/SpringFootballMall/blob/master/src/main/webapp/WEB-INF/views/admin/goods/modifyGoodsForm.jsp>

## 관리자 주문 관리 페이지

## 주문 조회

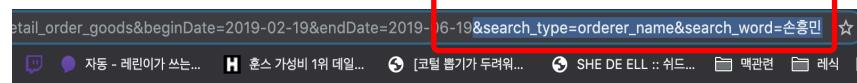
 간단조회 상세조회

2019  년 6  월 19  일 이전        까지 조회  
 조회 기간: 2019  년 02  월 19  일 ~ 2019  년 06  월 19   
 주문자이름

주문번호	주문일자	주문내역	배송상태	배송수정
450	2019-06-19	주문자:안지혁 주문자 번화번호:010-1111-11111 수령자:안지혁 수령자 번화번호:010-1111-11111 주문상품명(수량):파크 2 니트 쇼트 WB(725903-657)(725903657)(2) 주문상품명(수량):맨유 17-18 홀 쇼트(BQ3739)(1)	배송준비중 <input type="button" value="▼"/>	배송수정
449	2019-06-19	주문자:안지혁 주문자 번화번호:010-1111-11111 수령자:안지혁 수령자 번화번호:010-1111-11111 주문상품명(수량):파크 2 니트 쇼트 WB(725903-657)(725903657)(2) 주문상품명(수량):맨유 17-18 홀 쇼트(BQ3739)(1)	배송준비중 <input type="button" value="▼"/>	배송수정

# 관리자(adminOrderMain.jsp)

## 관리자 주문 관리 페이지 상세 조회



**주문 조회**

간단조회  상세조회

2019년 6월 19일 이전        까지 조회

조회 기간: 2019년 02월 19일 ~ 2019년 06월 19일

주문자이름

주문번호	주문일자	주문내역	배송상태	배송수정
451	2019-06-19	주문자:손흥민 주문자 번화번호:011-4321- 1234 수령자:손흥민 수령자 번화번호:011-4321- 1234 주문상품명(수량):프레데터 19.1 HG(EF8994)(1)	<input type="button" value="배송준비중"/>	<input type="button" value="배송수정"/>

1 2 3 4 5 6 7 8 9 10 [next](#)



**주문 조회**

간단조회  상세조회

2019년 6월 19일 이전        까지 조회

조회 기간: 2019년 02월 19일 ~ 2019년 06월 19일

주문자이름

주문번호	주문일자	주문내역	배송상태	배송수정
450	2019-06-19	주문자:안지현 주문자 번화번호:010-1111-1111 수령자:안지현 수령자 번화번호:010-1111-1111 주문상품명(수량):파크 2 뉴트 쇼트 WB(725903-657)(725903657)(2)	<input type="button" value="배송준비중"/>	<input type="button" value="배송수정"/>
449	2019-06-19	주문자:안지현 주문자 번화번호:010-1111-1111 수령자:안지현 수령자 번화번호:010-1111-1111 주문상품명(수량):파크 2 뉴트 쇼트 WB(725903-657)(725903657)(2)	<input type="button" value="배송준비중"/>	<input type="button" value="배송수정"/>

# 관리자(adminOrderMain.jsp)(2)

## 주문 상태에 따른 표 색상 변경

주문번호	주문일자	주문내역	배송상태	배송수정
451	2019-06-19	주문자:손흥민 주문자 번화번호:011-4321- 1234 수령자:손흥민 수령자 번화번호:011-4321- 1234 주문상품명(수량):프레데터 19.1 HG(EF8994)(1)	주문취소 ▾	배송수정

주문번호	주문일자	주문내역	배송상태	배송수정
451	2019-06-19	주문자:손흥민 주문자 번화번호:011-4321- 1234 수령자:손흥민 수령자 번화번호:011-4321- 1234 주문상품명(수량):프레데터 19.1 HG(EF8994)(1)	반품 ▾	배송수정

주문번호	주문일자	주문내역	배송상태	배송수정
451	2019-06-19	주문자:손흥민 주문자 번화번호:011-4321- 1234 수령자:손흥민 수령자 번화번호:011-4321- 1234 주문상품명(수량):프레데터 19.1 HG(EF8994)(1)	배송완료 ▾	배송수정

주문번호	주문일자	주문내역	배송상태	배송수정
451	2019-06-19	주문자:손흥민 주문자 번화번호:011-4321- 1234 수령자:손흥민 수령자 번화번호:011-4321- 1234 주문상품명(수량):프레데터 19.1 HG(EF8994)(1)	배송중 ▾	배송수정

## 주문자의 주문 내역에도 변경

### 최근주문내역

[더보기 >](#)

주문번호	주문일자	주문상품	주문상태	주문취소
451	2019-06-19	프레데터 19.1 HG(EF8994)/1개/270mm	배송중	주문취소

## 관리자 주문 관리 상세 페이지

## 1. 주문 상세 정보

주문번호	주문상품명	수량	주문금액	배송비	예상착립금	주문금액합계
451	 프레데터 19.1 HG(EF8994)	1개	220000원 (10% 할인)	0원	1500원	220000원

## 2. 배송지 정보

배송방법	일반택배
받으실 분	손흥민
휴대폰번호	011-4321- 1234
유선전화(선택)	02-111- 1111
주소	우편번호:13536 도로명 주소:경기 성남시 분당구 판교역로2번길 1 (백현동) [지번 주소:경기 성남시 분당구 백현동 582-6] 분당아파트 101동 1202호
배송 메시지	
선물 포장	no

## 주문고객

이름   
 핸드폰   
 이메일

배송방법	일반택배
받으실 분	손흥민
휴대폰번호	011-4321- 1234
유선전화(선택)	02-111- 1111
주소	우편번호:13536 도로명 주소:경기 성남시 분당구 판교역로2번길 1 (백현동) [지번 주소:경기 성남시 분당구 백현동 582-6] 분당아파트 101동 1202호
배송 메시지	
선물 포장	no

주문고객  
 이름   
 핸드폰   
 이메일

## 3. 결제 정보

결제방법	신용카드 카드사:삼성 할부개월수:일시불
결제카드	삼성
할부기간	일시불

## 3. 배송 상태

# 관리자(orderDetail.jsp)(2)

주문 상태 변경 시 동기식으로 주문 상태 데이터 전송

```
<script type="text/javascript">
function fn_modify_order_state(order_id){
    var s_delivery_state=document.getElementById("s_delivery_state");
    var index = s_delivery_state.selectedIndex; //선택한 옵션의 인덱스를 구합니다.
    var value = s_delivery_state[index].value; //선택한 옵션의 값을 구합니다.
    console.log("value: " +value);
    $.ajax({
        type : "post",
        async : false, //false인 경우 동기식으로 처리한다.
        url : "${contextPath}/admin/order/modifyDeliveryState.do",
        data : {
            order_id:order_id,
            'delivery_state':value
        },
        success : function(data, textStatus) {
            if(data.trim()=='mod_success'){
                alert("주문 정보를 수정했습니다.");
                location.href="${contextPath}/admin/order/orderDetail.do?order_id="+order_id;
            }else if(data.trim()=='failed'){
                alert("다시 시도해 주세요.");
            }
        },
        error : function(data, textStatus) {
            alert("에러가 발생했습니다." +data);
        },
        complete : function(data, textStatus) {
            //alert("작업을완료 했습니다");
        }
    }); //end ajax
}
</script>
```

## 관리자의 회원 관리 메인 화면

회원 조회

간단조회  상세조회

2019 년 6 월 19 일 이전      당일      1주      2주      1개월      2개월      3개월      4개월      까지 조회

조회 기간: 2019 년 02 월 19 일 ~ 2019 년 06 월 19 일      조회

✓ 회원이름      회원아이디      회원휴대폰번호      회원주소

		휴대폰번호	주소	가입일	탈퇴여부
korea02	손흥민	011-4321- 1234	경기 성남시 분당구 판교역로2번길 1 (백현동) 경기 성남시 분당구 백현동 582-6 분당아파트 101동 1202호	2019-06-07	활동중
admin	관리자	010-4197-9322	울산 남구 꽃대나리로 3 (달동) 울산 남구 달동 1297-5 101동 1604호	2019-06-02	활동중
korea01	안지혁	010-1111-11111	경기 성남시 분당구 판교역로 4 (백현동) 경기 성남시 분당구 백현동 582-7 우방아파트 101동 1604호	2019-05-27	활동중

## 관리자의 회원 관리 상세 페이지

## 손흥민님의 정보

아이디	korea02	<button>수정</button>
비밀번호	****	<button>수정</button>
이름	손흥민	<button>수정</button>
성별	<input checked="" type="radio"/> 여성 <input type="radio"/> 남성	<button>수정</button>
생년월일	1991 <input type="button" value="년"/> 3 <input type="button" value="월"/> 7 <input type="button" value="일"/>	<button>수정</button>
전화번호	02 <input type="button" value=" "/> -1111- <input type="button" value=" "/> 1111	<button>수정</button>
휴대폰번호	011 <input type="button" value=" "/> -4321- <input type="button" value=" "/> 1234	<button>수정</button>
<input checked="" type="checkbox"/> 쇼핑몰에서 발송하는 SMS 소식을 수신합니다.		
이메일(e-mail)	sonaldo @ hanmail.net	<input type="button" value="직접입력"/>
<input checked="" type="checkbox"/> 쇼핑몰에서 발송하는 e-mail을 수신합니다.		
<input type="text" value="13536"/> <a href="#">우편번호검색</a> 지번 주소: 경기 성남시 분당구 판교역로2번길 1 (백현동)		
주소	도로명 주소: 경기 성남시 분당구 백현동 582-6	
	나머지 주소: 분당아파트 101동 1202호	

회원탈퇴

## 탈퇴 처리 시 회원 상태 변경 됨

korea02 손흥민 011-4321- 1234 경기 성남시 분당구 판교역로2번길 1 (백현동) 경기 성남시 분당구 백현동 582-6 분당아파트 101동 1202호 2019-06-07 탈퇴

## 탈퇴 회원 복원 가능

주소	13536 <a href="#">우편번호검색</a>
	지번 주소: 경기 성남시 분당구 판교역로2번길 1 (백현동)
도로명 주소: 경기 성남시 분당구 백현동 582-6	
나머지 주소: 분당아파트 101동 1202호	

회원복원

## 회원 탈퇴,복원 버튼 변경 코드

```
<tr>
    <td>
        <input type="hidden" name="command" value="modify_my_info" />
        <c:choose>
            <c:when test="${member_info.delete_yn=='Y' }">
                <input class="btn btn-primary" type="button" value="회원복원" onClick="fn_delete_member('${member_info.member_id }','N')">
            </c:when>
            <c:when test="${member_info.delete_yn=='N' }">
                <input class="btn btn-primary" type="button" value="회원탈퇴" onClick="fn_delete_member('${member_info.member_id }','Y')">
            </c:when>
        </c:choose>
    </td>
</tr>
```

# 관리자(memberDetail.jsp)(3)

회원 정보 수정 버튼 클릭 시 동작 하는 자바  
스크립트 코드

```
<script>
function fn_modify_member_info(member_id,mod_type){
    var value;
    var frm_mod_member=document_frm_mod_member;
    if(mod_type=='member_pw'){
        value=frm_mod_member.member_pw.value;
    }else if(mod_type=='member_gender'){
        var member_gender=frm_mod_member.member_gender;
        for(var i=0; i<member_gender.length; i++){
            if(member_gender[i].checked){
                value=member_gender[i].value;
                break;
            }
        }
        //alert("member_gender111:"+value);
    }else if(mod_type=='member_birth'){
        var member_birth_y=frm_mod_member.member_birth_y;
        var member_birth_m=frm_mod_member.member_birth_m;
        var member_birth_d=frm_mod_member.member_birth_d;

        for(var i=0; i<member_birth_y.length;i++){
            if(member_birth_y[i].selected){
                value_y=member_birth_y[i].value;
                break;
            }
        }
        for(var i=0; i<member_birth_m.length;i++){
            if(member_birth_m[i].selected){
                value_m=member_birth_m[i].value;
                break;
            }
        }
        for(var i=0; i<member_birth_d.length;i++){
            if(member_birth_d[i].selected){
                value_d=member_birth_d[i].value;
                break;
            }
        }
        value+=value_y+"," +value_m+"," +value_d;
    }
}
```

```
}else if(mod_type=='tel'){
    var tel1=frm_mod_member.tel1,
    var tel2=frm_mod_member.tel2;
    var tel3=frm_mod_member.tel3;

    for(var i=0; tel1.length;i++){
        if(tel1[i].selected){
            value_tel1=tel1[i].value;
            break;
        }
    }
    value_tel2=tel2.value;
    value_tel3=tel3.value;

    value=value_tel1+" "+value_tel2+", "+value_tel3;
}else if(mod_type=='hp'){
    var hp1=frm_mod_member.hp1;
    var hp2=frm_mod_member.hp2;
    var hp3=frm_mod_member.hp3;
    var smssts_yn=frm_mod_member.smssts_yn;

    for(var i=0; i<hp1.length;i++){
        if(hp1[i].selected){
            value_hp1=hp1[i].value;
            break;
        }
    }
    value_hp2=hp2.value;
    value_hp3=hp3.value;
    value_smssts_yn=smssts_yn.checked;

    value=value_hp1+" "+value_hp2+", "+value_hp3+","+value_smssts_yn;
```

해당하는 요소의 태입을 키  
로 받아 각 하나의 항목 씩  
설정

:

회원 정보 수정 버튼 클릭 시 동작 하는 자바  
스크립트 코드

```
$.ajax({
    type : "post",
    async : false, //false인 경우 동기식으로 처리한다.
    url : "${contextPath}/admin/member/modifyMemberInfo.do",
    data : {
        member_id:member_id,
        mod_type:mod_type,
        value:value
    },
    success : function(data, textStatus) {
        if(data.trim()=='mod_success'){
            alert("회원 정보를 수정했습니다.");
        }else if(data.trim()=='failed'){
            alert("다시 시도해 주세요.");
        }
    },
    error : function(data, textStatus) {
        alert("에러가 발생했습니다." +data);
    },
    complete : function(data, textStatus) {
        //alert("작업을완료 했습니다");
        document.location.href="${contextPath}/admin/member/memberDetail.do?member_id=${member_info.member_id}";
    }
}); //end ajax
```

동기식으로 회원의 아이디,  
요소의 항목 키, 값을 전송

수정 후 다시 현재 페이지  
로 이동 해서 변경 되었음  
을 확인(get방식)

# Index



01

서론

프로젝트 목적

프로젝트 개요

02

본론

프로젝트 구성도

프로젝트 기능

프로젝트 주요 기술

03

결론

개선 사항

프로젝트 후기

- ORM을 Mybatis에서 Hibernate로 교체
- 이전 게시판 개발에서 사용한 페이징 처리로 교체
- Spring Boot를 활용해서 개발
- Restful API로 변경

- 기본적인 쇼핑몰 개념을 이해
- AOP, DI의 개념 이해
- 프레임 워크의 많은 장점 습득
- Spring의 3-layer 구조를 이해
- Spring에서 제공 하는 클래스들이 매우 유용함
- Mybatis를 이용해서 DB를 연동하는 것의 편리함



# Thank you

안지혁