



스프링으로 구현한 쇼핑몰 웹 개발

진명국

목차

-- 주요 기능 소개

---- 전체적인 개발환경

---- 개발 환경1

---- 개발 환경2

---- 개발 환경3

---- 웹 서버 동작 1

---- 웹 서버 동작 2

---- 웹 서버 동작 3

---- ERD 구조 1

---- ERD 구조 2

---- BookShop 설계도

---- 구성 요소

--비지니스 로직

-- 메인 페이지

---- 화면 요소

---- 빠른 메뉴

-- 회원 기능

---- 로그인

---- 회원 가입

-- 상품 기능

---- 상품 검색

-- 장바구니 기능

---- 장바구니 상품 추가

---- 장바구니 상품 조회

---- 장바구니 상품 수정

-- 주문 기능

---- 상품 주문 조회

---- 주문 상품 구매 (payUp)

-- 마이페이지 기능

---- 주문 내역 조회

---- 회원 정보 수정

---- 회원 탈퇴

전체적인 개발환경

Environment



FrameWork



BuildTool



DBMS



Payment



개발환경 1

Environment



Eclipse IDE

-이클립스를 사용한 개발환경

JavaServerPage

-접근된 JSP를 WAS는 jasper를 이용하여 ServletFile로 변환, Servlet을 실행하여 동적인 부분이 처리된 Html코드를 반환

JavaScript

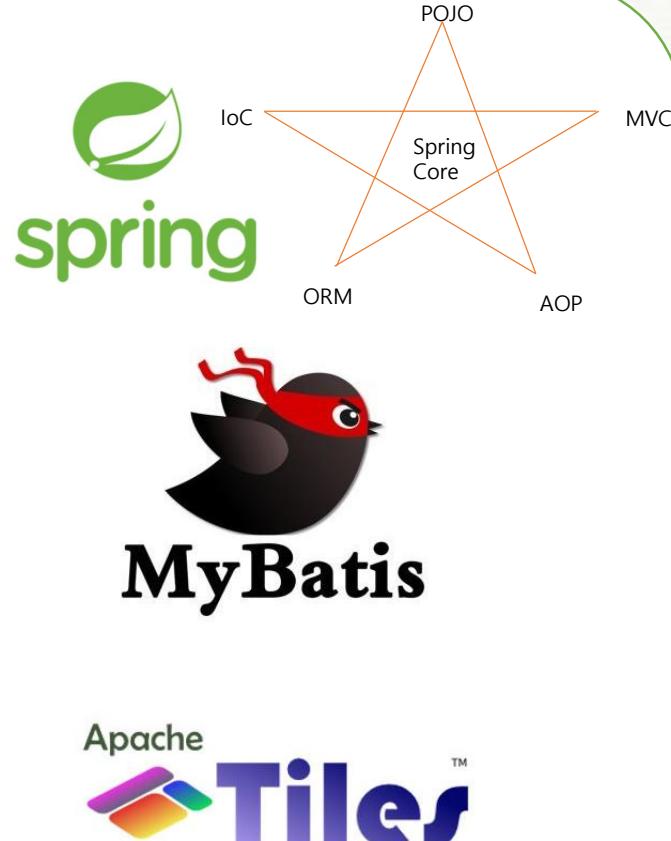
-Document Object Model을 제어할 수 있게 하는 Jquery
-비동기 방식으로 서버에 데이터를 요청하는 Asynchronous Javascript And Xml

Apache, Tomcat

-Client와 Request, Response로 정적 데이터를 처리하는 Apache
-Apache에서 AJP Protocol을 사용하여 WAS에게 동적 데이터를 요청하여 전달

개발환경 2

FrameWork



Spring

- POJO : 오래된 자바의 형식
- ORM : MyBatis Framework(영속성) 관련 연동 기능을 제공
- IoC : Xml 설정 또는 @Autowired의 DI 방식
- AOP : CoreConcern과 CrossCuttingConcern을 분리하는 Log4j, proxy
 @Aspect = PointCut + Advice
- MVC : Model And View Controller

MyBatis

- SQL실행 결과를 Beans, Map 객체에 Mapping해 주는 영속성 솔루션
- Mapper.xml에 정보를 저장하고 <![CDATA[]]>CRUD를 구현
- SQL문과 프로그래밍 코드를 분리해서 구현

Tiles

- page를 header, aside, body 등으로 구분하여
 가독성 및 효과으로 page를 관리하는 FrameWork

개발환경 3

BuildTool



DBMS



Payment



Maven

- pom.xml에 정의하여 Library를 네트워크를 통해 관리
- 중앙 저장소를 통한 자동 의존성 관리 BuildTool
- 간단하며 설정을 통한 배포 가능

OracleDBMS

- Structured Query Language, 스키마를 통한 저장소
- 기본적으로 select, insert, update, delete문이 대표적
- sqldeveloper툴을 사용하여 sqlplus를 편리하게 사용 가능

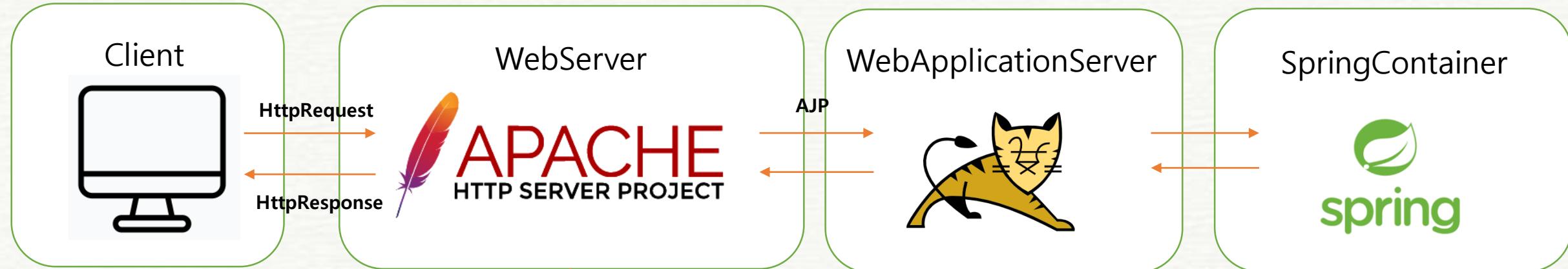
Payup

- 2차 PG사의 기업으로 1차 PG사에게 승인을 받은 값으로 카드사와 연결하여 결제를 진행하고 승인 및 거래 취소 등을 관리

웹 서버 동작 1

Apache + WAS의 장점

- 서버의 전체적인 부하를 분산
- 속도 향상
- Apache JServ Protocol



<WebServer>

정적 데이터 처리 서버

-이미지, html파일 같은 리소스만을 제공하는 서버는 웹 서버만 사용하여 빠르고 안정적이게 활용

-웹 서버와 WAS를 연동하여 각각의 역할을 분담시켜 요청에 대한 응답을 빠르고 효율적으로 처리

-Apache WebServer는 톰캣으로 전달 받은 처리 결과를 사용자에게 전송

<WebApplicationServer>

동적 데이터 처리 서버

-JSP, Servlet구동 환경(java)제공, Servlet관리

-DB로 연결되어 데이터를 주고 받거나 자바등을 통해 데이터 조작이 필요한 경우 WAS를 활용

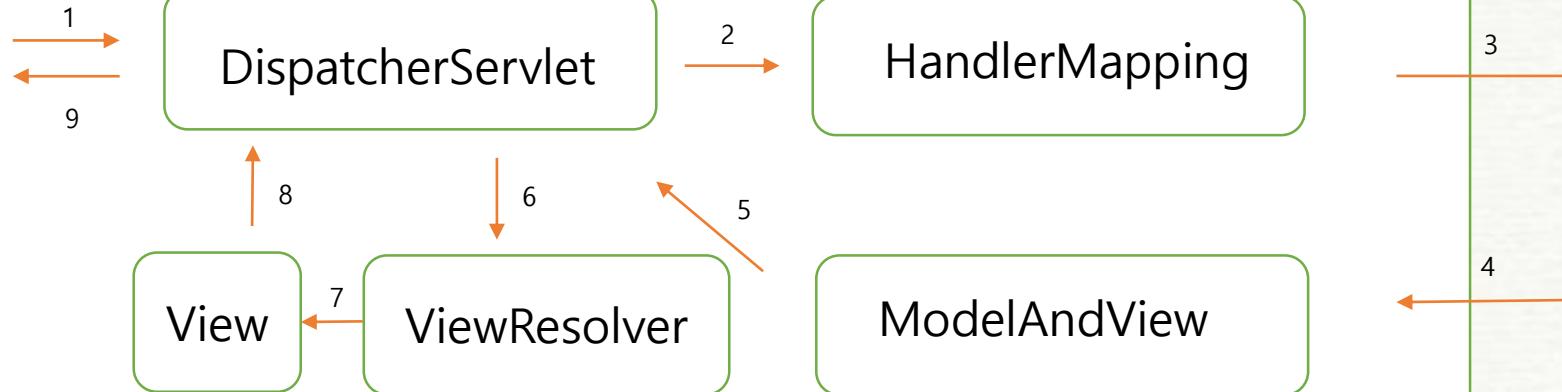
-Apache JServ Protocol port에 접속하여 정보를 주고 받음

웹 서버 동작 2

Spring MVC 패턴

Business Logic

Client



1. Client가 DispatcherServlet에 URL로 접근하여 정보를 요청
2. HandlerMapping에 매핑된 컨트롤러가 있는지 요청
3. 매핑된 컨트롤러에 대해 처리를 요청
4. 컨트롤러가 클라이언트의 요청을 처리한 결과와 View 이름을 ModelAndView에 저장
5. 저장된 결과를 Dispatcher로 보냄
6. Dispatcher에서는 컨트롤러에서 보내온 View 이름을 ViewResolver로 보내 해당 View를 요청
7. ViewResolver는 요청한 View를 보냄
8. View의 처리 결과를 Dispatcher로 보냄
9. Dispatcher의 최종적인 결과를 브라우저로 전송

Controller

MemberController.java
 MemberControllerImpl.java

Service@transactional

MemberService.java
 MemberServiceImpl.java

Repository@CRUD

MemberDAO.java
 MemberDAOImpl.java

DB

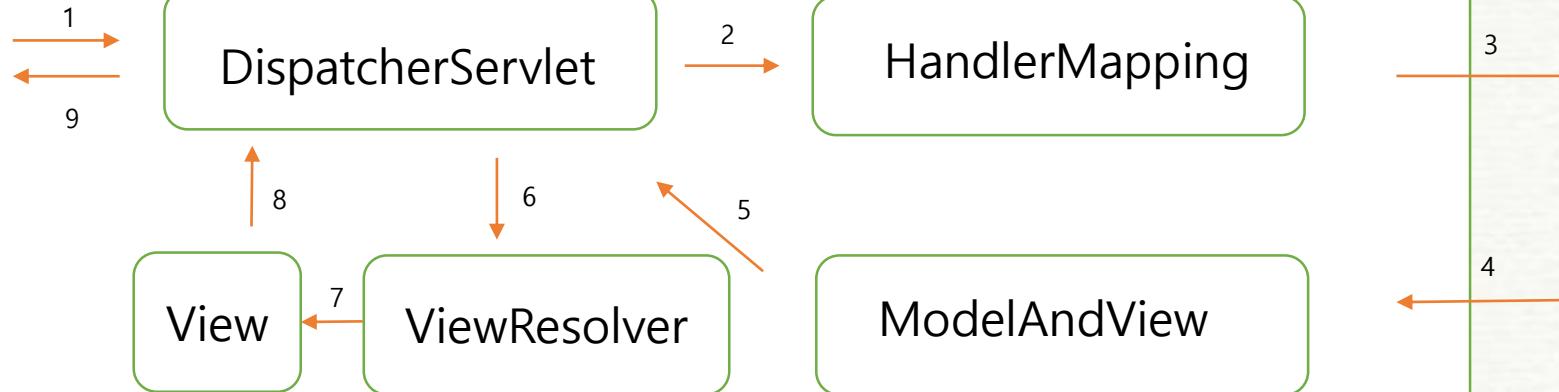


웹 서버 동작 3

Spring MCV

Business Logic

Client



Controller

MemberController.java
 MemberControllerImpl.java

Service@transactional

MemberService.java
 MemberServiceImpl.java

Repository@CRUD

MemberDAO.java
 MemberDAOImpl.java

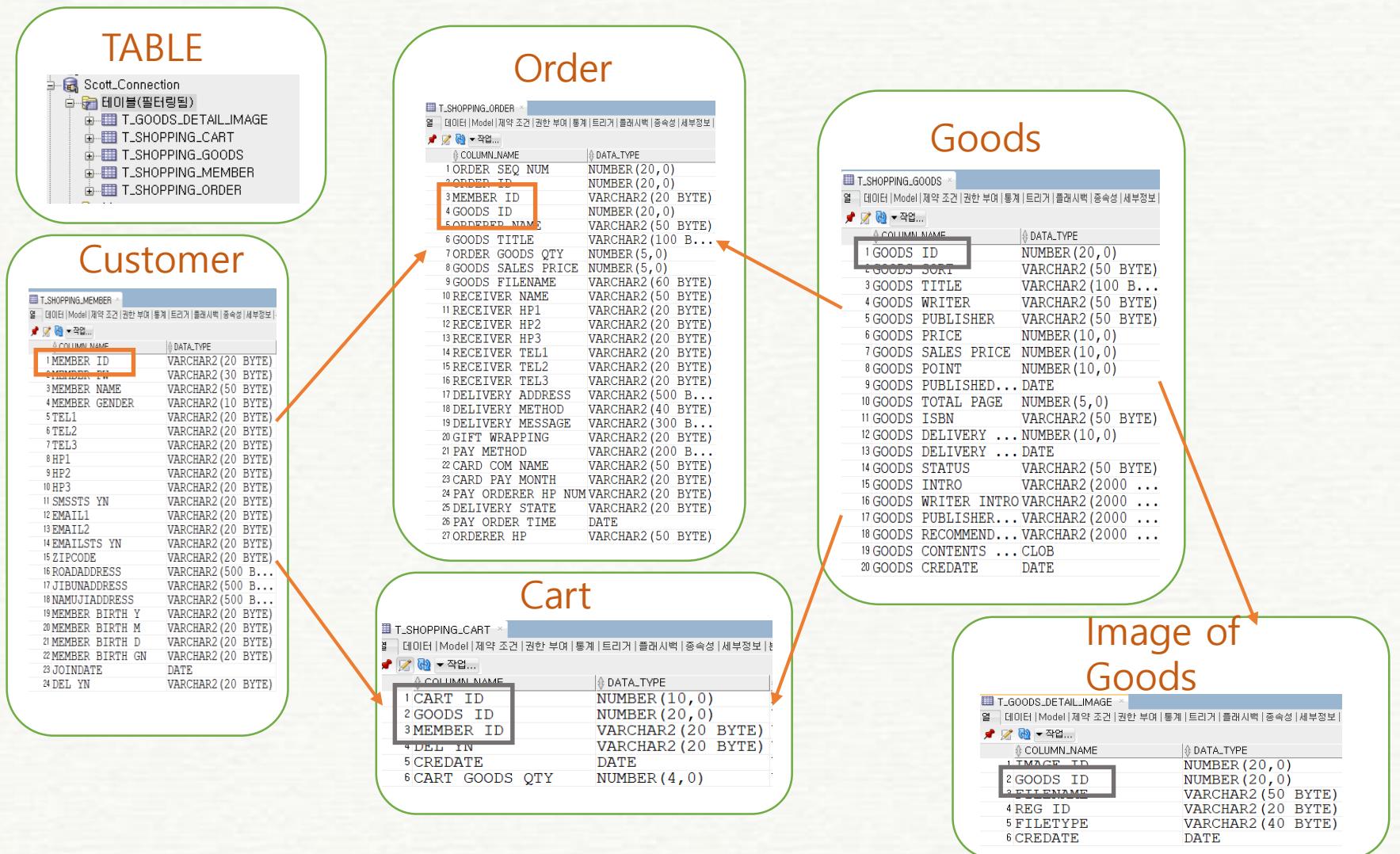
DB



MVC패턴을 사용해야하는 이유

- Business Logic과 UI로직을 분리하여 유지보수를 독립적으로 수행
- 보수성, 유연성, 확장성을 보장해서 중복 코딩 문제 완화
- 쉽고 빠르게 개발 가능

ERD 구조 1



DB

Entity Relationship Diagram

테이블을 따로 구성하고
고유값인 primary key를
사용하여 각각의
Schema를 연결

ERD 구조 2

```
select * from t_shopping_member m, t_shopping_cart c  
where m.member_id = c.member_id;
```

질의 결과 x

SQL | 인출된 모든 행: 9(0,004초)

	MEMB...	GOODS_ID
1	hi	Y	j	..	Y	5	..	2	..	N	1	354	..	N	..	1						
2	hi	Y	j	..	Y	5	..	2	..	N	2	336	..	N	..	1						
3	hi	Y	j	..	Y	5	..	2	..	N	3	342	..	N	..	1						
4	hello	Y	Y	5	..	2	..	N	4	340	..	N	..	1							
5	hello	Y	Y	5	..	2	..	N	5	338	..	N	..	1							
6	hello	Y	Y	5	..	2	..	N	6	341	..	N	..	1							
7	nice	Y	Y	5	..	2	..	N	7	337	..	N	..	1							
8	nice	Y	Y	5	..	2	..	N	8	354	..	N	..	1							
9	nice	Y	Y	5	..	2	..	N	9	336	..	N	..	1							



DB

Entity Relationship Diagram

-member 스키마와
cart스키마의 key값을
연결하여 조회하는 기능

-key와 value 값으로
query문을 작성해서
DB에 적재

BookShop 설계도

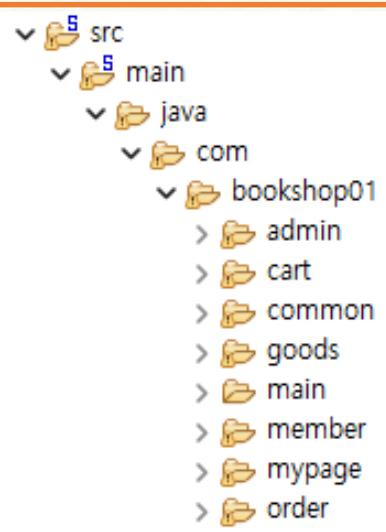
기능 세부기능 VO mapper @RequestMapping 컨트롤러 파일 컨트롤러 기능 컨트롤러 매개변수 컨트롤러 리턴타입

기능	세부 기능	VO	mapper (서비스 impl 파일)	@RequestMapping(from controller.java)	컨트롤러 파일	컨트롤러 기능(메서드)	컨트롤러 매개변수	컨트롤러 매개변수	컨트롤러 리턴타입
메인 페이지 (src/main/java/main)	메인 페이지 빠른 메뉴	없음	없음	/main/main.do	MainController	main	일반 매개변수 없음	HttpServletRequest request, HttpServletResponse response	ModelAndView
회원 기능 (src/main/java/member)	로그인	MemberVO	memberDAO.login() memberDAO.insertNewMember() memberDAO.selectOverlappedID()	/member /member/login.do /member/logout.do /member/addMember.do /member/overlapped.do	MemberController	login logout addMember overlapped	@RequestParam Map<String, String> loginMap, 일반 매개변수 없음 @ModelAttribute("memberVO") MemberVO _memberVO, @RequestParam("id") String id,	HttpServletRequest request, HttpServletResponse response	ModelAndView ModelAndView ResponseEntity ResponseEntity
	회원가입								
상품 기능 (src/main/java/goods)	상품 검색	GoodsVO, ImageFileVO	goodsDAO.selectGoodsList() goodsDAO.selectGoodsDetail() goodsDAO.selectGoodsDetailImage() goodsDAO.selectKeywordSearch() goodsDAO.selectGoodsBySearchWord()	/goods /goods/goodsDetail.do /goods/keywordSearch.do /goods/searchGoods.do	GoodsController	goodsDetail keywordSearch searchGoods addGoodsInQuick	@RequestParam("goods_id") String goods_id, @RequestParam("keyword") String keyword, @RequestParam("searchWord") String searchWord, String goods_id, GoodsVO goodsVO	HttpServletRequest request, HttpServletResponse response HttpServletRequest request, HttpServletResponse response HttpServletRequest request, HttpServletResponse response HttpSession session	ModelAndView @ResponseBody ModelAndView void
장바구니 기능 (src/main/java/cart)	장바구니 상품 조회	CartVO	cartDAO.selectCartList() cartDAO.selectGoodsList() cartDAO.selectCountInCart() cartDAO.insertGoodsInCart() cartDAO.updateCartGoodsQty() cartDAO.deleteCartGoods()	/cart /cart/myCartList.do /cart/addGoodsInCart.do /cart/modifyCartQty.do /cart/removeCartGoods.do	CartController	myCartMain	일반 매개변수 없음	HttpServletRequest request, HttpServletResponse response	ModelAndView @ResponseBody @ResponseBody ModelAndView
	장바구니 상품 추가					addGoodsInCart	@RequestParam("goods_id") int goods_id		
	장바구니 상품 수정					modifyCartQty	@RequestParam("goods_id") int goods_id @RequestParam("cart_goods_qty") int cart_goods_qty.		
	장바구니 상품 삭제					removeCartGoods	@RequestParam("cart_id") int cart_id		
주문 기능 (src/main/java/order)	상품 주문	OrderVO	orderDAO.listMyOrderGoods() orderDAO.insertNewOrder() orderDAO.removeGoodsFromCart() orderDAO.findMyOrder()	/order /order/orderEachGoods.do /order/orderAllCartGoods.do /order/payToOrderGoods.do	OrderController	orderEachGoods	@ModelAttribute("orderVO") OrderVO _orderVO,	HttpServletRequest request, HttpServletResponse response	ModelAndView ModelAndView ModelAndView
	주문 상품 조회					orderAllCartGoods	@RequestParam("cart_goods_qty") String[] cart_goods_qty,		
	주문 상품 취소					payToOrderGoods	@RequestParam Map<String, String> receiverMap,		
마이페이지 기능 (src/main/java/mypage)	마이페이지 메인	MyPageVO	myPageDAO.selectMyOrderGoodsList() myPageDAO.selectMyOrderInfo() myPageDAO.selectMyOrderHistoryList() myPageDAO.selectMyDetailInfo() myPageDAO.updateMyOrderCancel() myPageDAO.selectMyDetailInfo()	/mypage /mypage/myPageMain.do /mypage/myOrderDetail.do /mypage/listMyOrderHistory.do /mypage/cancelMyOrder.do /mypage/myDetailInfo.do /mypage/modifyMyInfo.do	MyPageController	myPageMain	@RequestParam(required = false,value="message") String message,	HttpServletRequest request, HttpServletResponse response	ModelAndView
	주문 내역 리스트 조회					myOrderDetail	@RequestParam("order_id") String order_id,		ModelAndView
	주문 내역 상세					listMyOrderHistory	@RequestParam Map<String, String> dateMap,		ModelAndView
	회원 정보 상세					cancelMyOrder	@RequestParam("order_id") String order_id,		ModelAndView
	회원 정보 수정					myDetailInfo	일반 매개변수 없음		ModelAndView
	회원 탈퇴					modifyMyInfo	@RequestParam("attribute") String attribute, @RequestParam("value") String value		ResponseEntity

▶ 구성 요소

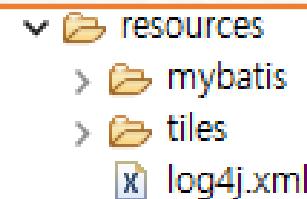
Main.java

- admin
- cart
- common
- goods
- main
- member
- mypage
- order



Main.resources

- mybatis
- tiles



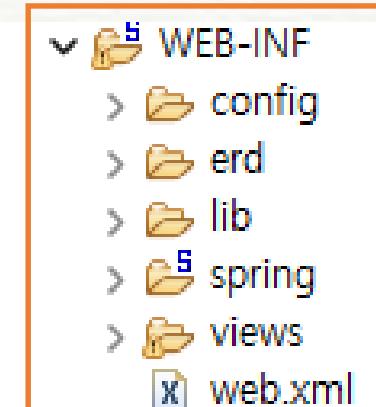
Webapp.resources

- CSS
- image
- jquery

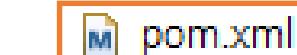


webapp.WEB-INF

- config
- erd
- lib
- spring
- view
- web.xml



pom.xml



서비스 환경

--비지니스 로직

-- 메인 페이지

---- 화면 요소

---- 빠른 메뉴

-- 회원 기능

---- 로그인

---- 회원 가입

-- 상품 기능

---- 상품 검색

-- 장바구니 기능

---- 장바구니 상품 추가

---- 장바구니 상품 조회

---- 장바구니 상품 수정

-- 주문 기능

---- 상품 주문 조회

---- 주문 상품 구매(payUp)

-- 마이페이지 기능

---- 주문 내역 조회

---- 회원 정보 수정

---- 회원 탈퇴

Business Logic

Controller

MemberController.java
 MemberControllerImpl.java

Service@transactional

MemberService.java
 MemberServiceImpl.java

Repository@CRUD

MemberDAO.java
 MemberDAOImpl.java

DB



기능

메인페이지

세부기능

화면 요소

빠른 메뉴

회원 기능

로그인

회원 가입

상품 기능

상품 검색

장바구니 기능

장바구니 상품 추가

장바구니 상품 조회

장바구니 상품 수정

주문 기능

주문 상품 조회

주문 상품 구매(payUp)

마이페이지 기능

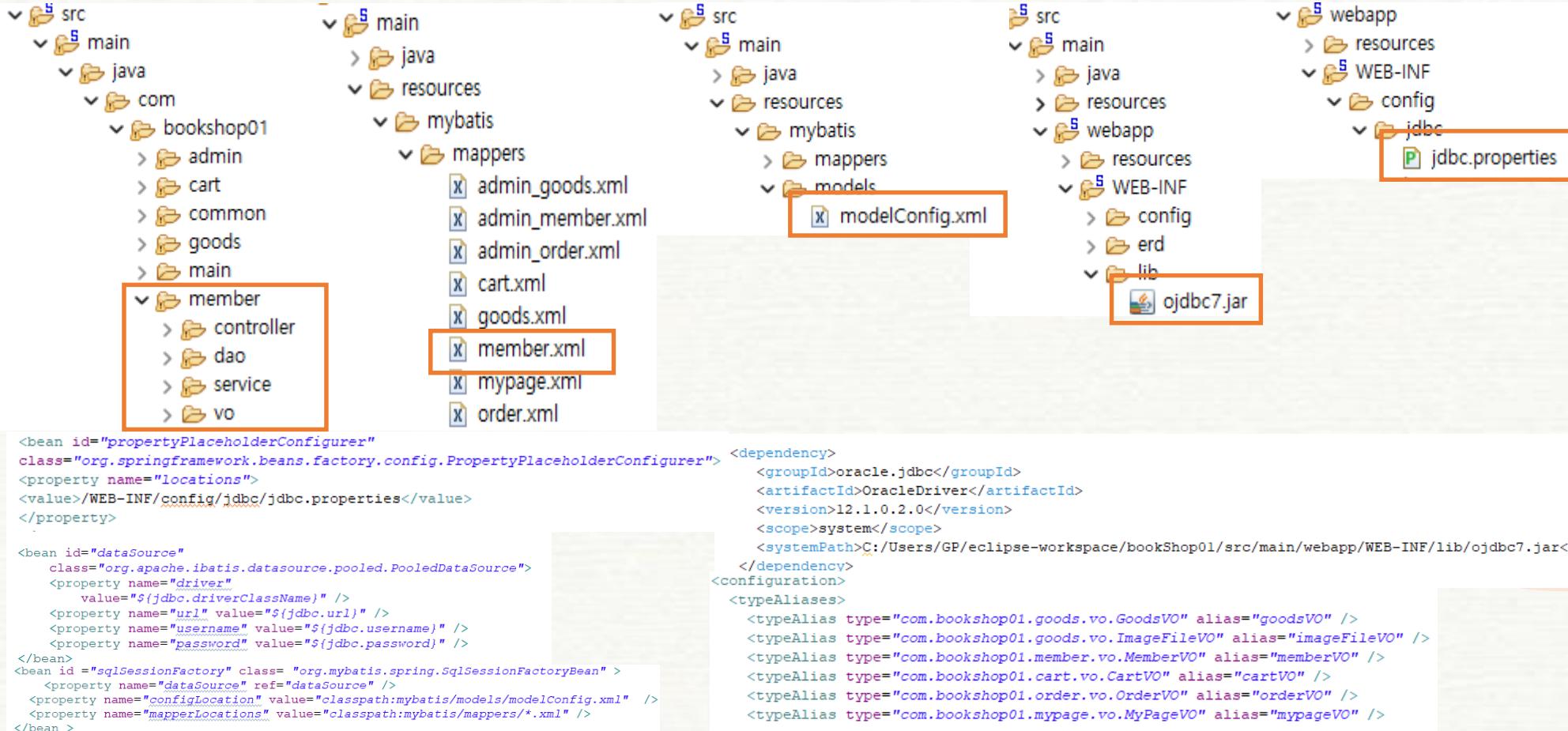
주문 내역 조회

회원 정보 수정

회원 탈퇴

비지니스 로직

MVC / mybatis / container / DB.OracleSQL



웹 페이지의 기능을 알아보기 위해 **비지니스 로직**의 MVC패턴과 mybatis, container, DB의 연관 관계와 그 기능들이 어떻게 동작하는지 알아보겠습니다.

비지니스 로직

Business Logic

Controller

MemberController.java
MemberControllerImpl.java



Service@Transactional

MemberService.java
MemberServiceImpl.java



Repository@CRUD

MemberDAO.java
MemberDAOImpl.java



interface MemberController

```
public interface MemberController {  
    public ModelAndView login(@RequestParam Map<String, String> loginMap, HttpServletRequest request, HttpServletResponse response) throws Exception;  
    public ModelAndView logout(HttpServletRequest request, HttpServletResponse response) throws Exception;  
    public ResponseEntity addMember(@ModelAttribute("member") MemberVO member, HttpServletRequest request, HttpServletResponse response) throws Exception;  
    public ResponseEntity overlapped(@RequestParam("id") String id, HttpServletRequest request, HttpServletResponse response) throws Exception;  
}
```

inherit

```
@Controller("memberController")  
class MemberControllerImpl
```

```
@Controller("memberController")  
@RequestMapping(value="/member")  
public class MemberControllerImpl extends BaseController implements MemberController {  
    @Autowired  
    private MemberService memberService;  
    @Autowired  
    private MemberVO memberVO;
```

- MemberController는 MemberControllerImpl를 상속해 interface를 기능을합니다.
- **왜 상속을 받아야 하는가** 하는 의문이 생기는데 **Open Closed Principle**의 원칙에 의해 확장에는 열려있고 수정에는 닫혀 있어야 한다는 프로그래밍의 원칙 때문입니다

비지니스 로직

Business Logic

Controller

MemberController.java
MemberControllerImpl.java



Service@transactional

MemberService.java
MemberServiceImpl.java



Repository@CRUD

MemberDAO.java
MemberDAOImpl.java



@Component

는
@Controller, @Service, @Repository 어노테이션의 구체화 된 형태
로 세분화 된 어노테이션들은 가독성을 높여줍니다.

```
@Controller("memberController")
@RequestMapping(value="/member")
public class MemberControllerImpl extends BaseController implements MemberController{
    @Autowired
    private MemberService memberService;
    @Autowired
    private MemberVO memberVO;

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

@Repository("memberDAO")
public class MemberDAOImpl implements MemberDAO{
    @Autowired
    private SqlSession sqlSession;
    <bean id = "sqlSession"
        class= "org.mybatis.spring.SqlSessionTemplate" >
        <constructor-arg ref= "sqlSessionFactory"></constructor-arg >
    </bean >

    Scott_Connection
    테이블(필터링됨)
    T_GOODS_DETAIL_IMAGE
    T_SHOPPING_CART
    T_SHOPPING_GOODS
    T_SHOPPING_MEMBER
```

@Controller는 중간 제어자 역할을 수행하며 사용자의 요청을 처리 한 후 지정된 View를 반환합니다.

@Service는 Repository와 Controller의 중간에서 transactional의 기능을 수행합니다.

@Repository는 mybatis.mapper와 연결하여 CRUD 문법을 수행합니다.

비지니스 로직

Business Logic

Controller

MemberController.java
MemberControllerImpl.java



Service@transactional

MemberService.java
MemberServiceImpl.java



Repository@CRUD

MemberDAO.java
MemberDAOImpl.java



DB
SQL

```
@Controller("memberController")
@RequestMapping(value="/member")
public class MemberControllerImpl extends BaseController implements MemberController{
    @Autowired
    private MemberService memberService;
    @Autowired
    private MemberVO memberVO;

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

        @Repository("memberDAO")
        public class MemberDAOImpl implements MemberDAO{
            @Autowired
            private SqlSession sqlSession;
        }
    }
}

<bean id="sqlSession"
      class="org.mybatis.spring.SqlSessionTemplate" >
    <constructor-arg ref="sqlSessionFactory"></constructor-arg >
</bean >

Scott_Connection
├── 테이블(필터링됨)
│   +-- T_GOODS_DETAIL_IMAGE
│   +-- T_SHOPPING_CART
│   +-- T_SHOPPING_GOODS
│   +-- T_SHOPPING_MEMBER
```

@Autowired는

각 상황의 타입에 맞는 IoC컨테이너 안에 존재하는 bean을 자동으로 주입해 줍니다.

Autowired 어노테이션을 사용하면 객체의 의존성을 가지는 부분에 간단히 의존성 주입을 받을 수 있습니다.

비지니스 로직

webapp / WEB-INF / web.xml
ContextLoaderListener

```
<listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>
```

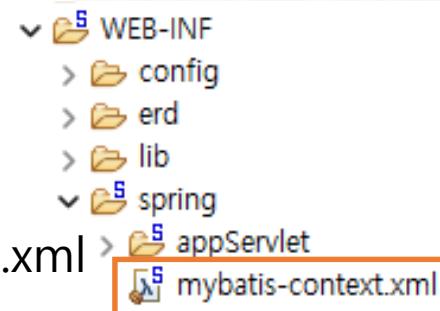
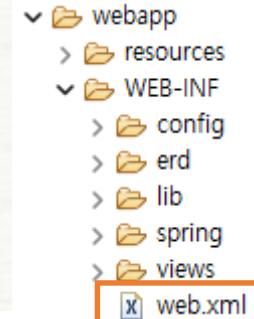
contextConfigLacation

```
<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/spring/mybatis-context.xml</param-value>
</context-param>
```

webapp / WEB-INF / spring / mybatis-context.xml
SqlSessionTemplate

```
<bean id = "sqlSession"
      class= "org.mybatis.spring.SqlSessionTemplate" >
    <constructor-arg ref= "sqlSessionFactory"></constructor-arg >
</bean >
```

```
<bean id="propertyPlaceholderConfigurer"
      class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
    <property name="locations">
      <value>/WEB-INF/config/jdbc/jdbc.properties</value>
    </property>
</bean>
```



ContextLoaderListener

- WAS 구동 시 web.xml을 읽어들여 webApplication 설정을 구성, 초기 세팅 작업을 해주는 역할
- ServletContainer 생명주기에 맞춰서 spring의 application context를 servletAttribute에 등록, 제거

contextConfigLacation

- springFrameWork가 동작하기 위한 설정 파일의 위치를 알려주는 파라미터

mybatis-context.xml 실행

- sqlSessionFactory에 접근

비지니스 로직

webapp / WEB-INF / web.xml
ContextLoaderListener

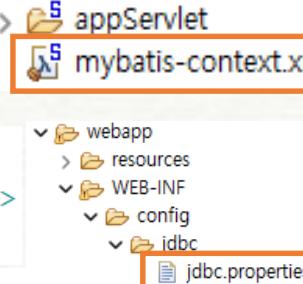
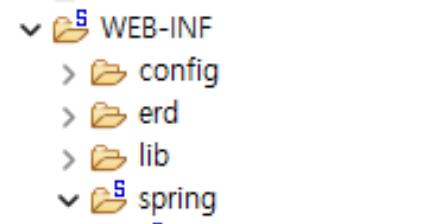
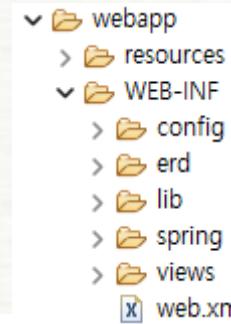
```
<listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>
```

```
<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/spring/mybatis-context.xml</param-value>
</context-param>
```

webapp / WEB-INF / spring / mybatis-context.xml
SqlSessionTemplate

```
<bean id="sqlSession"
      class="org.mybatis.spring.SqlSessionTemplate">
    <constructor-arg ref="sqlSessionFactory"></constructor-arg>
</bean>
```

```
<bean id="propertyPlaceholderConfigurer"
      class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
    <property name="locations">
      <value>/WEB-INF/config/jdbc/jdbc.properties</value>
    </property>
</bean>
```



SqlSessionTemplate

- mybatis, spring 연동의 핵심
- mybatis에서 sqlSession을 생성하기 위해 sqlSessionFactory를 사용
- 세션을 한번 생성하면 매핑 구문을 실행하고 더이상 필요하지 않은 상태가 되면 세션 종료
- 커밋하거나 롤백하는 것을 포함한 세션의 생명 주기를 관리
- propertyPlaceholderConfigurer은 jdbc.properties를 사용하기 위한 설정

비지니스 로직



mybatis-contents.xml

```
<bean id="dataSource"
      class="org.apache.ibatis.datasource.pooled.PooledDataSource">
    <property name="driver"
              value="${jdbc.driverClassName}" />
    <property name="url" value="${jdbc.url}" />
    <property name="username" value="${jdbc.username}" />
    <property name="password" value="${jdbc.password}" />
</bean>
```

mybatis-contents.xml

```
<dependency>
  <groupId>oracle.jdbc</groupId>
  <artifactId>OracleDriver</artifactId>
  <version>12.1.0.2.0</version>
  <scope>system</scope>
  <systemPath>C:/Users/GP/eclipse-workspace/bookShop01/src/main/webapp/WEB-INF/lib/ojdbc7.jar</systemPath>
</dependency>
```

pom.xml

```
1 jdbc.driverClassName=oracle.jdbc.driver.OracleDriver
2 jdbc.url=jdbc:oracle:thin:@localhost:1521:orcl
3 jdbc.username=scott
4 jdbc.password=12341234
```

```
</bean>
<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>
```

COLUMNNAME	DATA TYPE
1 MEMBER ID	VARCHAR2(20)
2 MEMBER PW	VARCHAR2(30)
3 MEMBER NAME	VARCHAR2(50)
4 MEMBER GENDER	VARCHAR2(10)
5 TELL1	VARCHAR2(20)
6 TELL2	VARCHAR2(20)
7 TELL3	VARCHAR2(20)
8 HP1	VARCHAR2(20)
9 HP2	VARCHAR2(20)
10 HP3	VARCHAR2(20)
11 SMSSTS YN	VARCHAR2(20)
12 EMAIL1	VARCHAR2(20)
13 EMAIL2	VARCHAR2(20)
14 EMAILSTS YN	VARCHAR2(20)
15 ZIPCODE	VARCHAR2(20)
16 ROADADDRESS	VARCHAR2(50)
17 JIBUNADDRESS	VARCHAR2(50)
18 NAMUJIADDRESS	VARCHAR2(50)
19 MEMBER BIRTH Y	VARCHAR2(20)
20 MEMBER BIRTH M	VARCHAR2(20)
21 MEMBER BIRTH D	VARCHAR2(20)
22 MEMBER BIRTH GN	VARCHAR2(20)
23 JOINDATE	DATE
24 DEL YN	VARCHAR2(20)

접속 정보 - [Scott_Connection]

사용자 이름(U):	scott
비밀번호(P):	*****

도움말(H) 확인 취소

sqlSessionFactory

- 속성은 **dataSource**로 DB의 객체를 조회
- { driver : jdbc.driverClassName }
- { url : jdbc.url }
- { username : jdbc.username }
- { password : jdbc.password }

비지니스 로직



mybatis-context.xml

```
<bean id="sqlSession"
      class="org.mybatis.spring.SqlSessionTemplate">
    <constructor-arg ref="sqlSessionFactory"/>
</bean>

@Repository("memberDAO")
public class MemberDAOImpl implements MemberDAO{
    @Autowired
    private SqlSession sqlSession;

    @Override
    public MemberVO login(Map loginMap) throws DataAccessException{
        MemberVO member=(MemberVO)sqlSession.selectOne("mapper.member.login",loginMap);
        return member;
    }
}
```

```
</bean>
<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>
```

member.xml

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

<mapper namespace="mapper.member">
    <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="member_birth_y" column="member_birth_y" />
        <result property="member_birth_m" column="member_birth_m" />
        <result property="member_birth_d" column="member_birth_d" />
    </resultMap>
    <select id="login" parameterType="Map" resultType="MemberVO">
        select * from member
        where member_id = #{member_id}
    </select>
</mapper>
```

sqlSession

- DAO와 연동
- MemberVO(Map loginMap) / VO객체를 Map으로 저장
- MemberVO **member** =
(MemberVO) sqlSession.selectOne(mapper.member.login)

mapperLocations

- mappers/member.xml과 연동
- MemberVO와 연동
- <![CDATA[
select * from member
where member_id = #{member_id}]]>

비지니스 로직



```
</bean>
<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 >
```

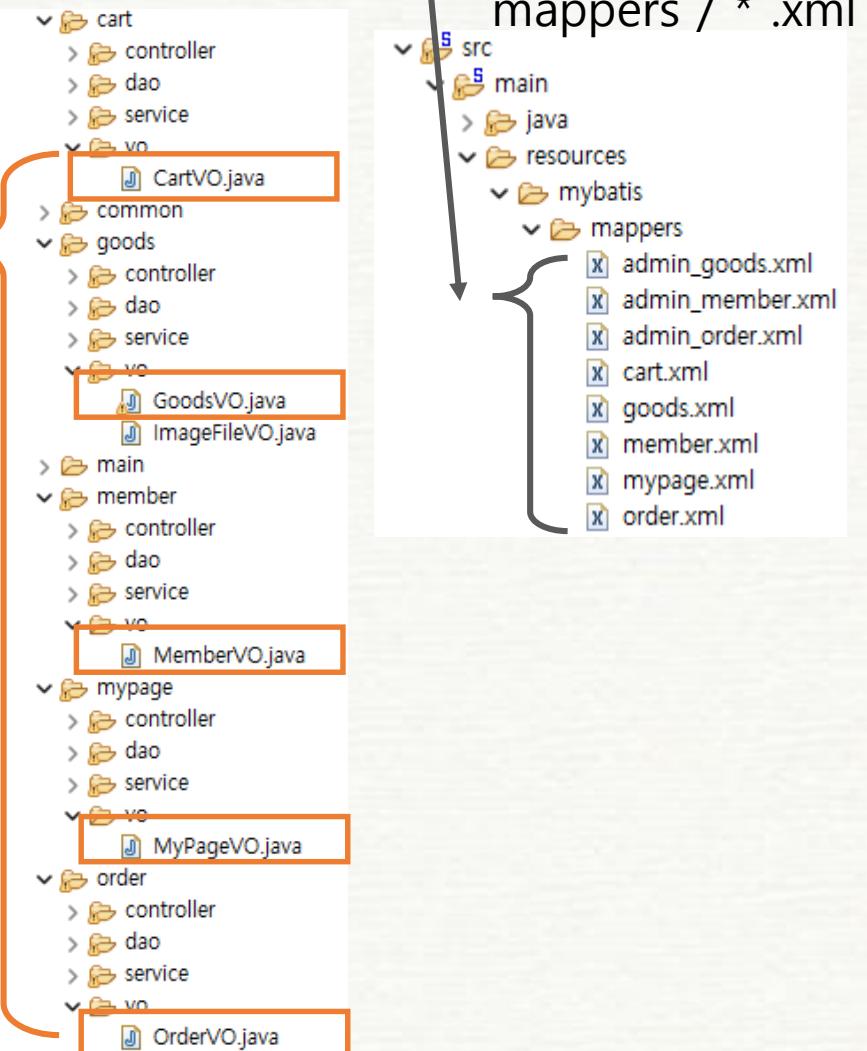
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-config.dtd">

<configuration>
    <typeAliases>
        <typeAlias type="com.bookshop01.goods.vo.GoodsVO" alias="goodsVO" />
        <typeAlias type="com.bookshop01.goods.vo.ImageFileVO" alias="imageFileVO" />
        <typeAlias type="com.bookshop01.member.vo.MemberVO" alias="memberVO" />
        <typeAlias type="com.bookshop01.cart.vo.CartVO" alias="cartVO" />
        <typeAlias type="com.bookshop01.order.vo.OrderVO" alias="orderVO" />
        <typeAlias type="com.bookshop01.mypage.vo.MyPageVO" alias="mypageVO" />
    </typeAliases>
</configuration>
```

modelConfig.xml

configLocation

- mybatis의 기능인 **<typeAliases>** 태그를 사용
- modelConfig.xml은 com.bookshop01.....VO를 각각의 VO로 등록함으로 **class명을 간략하게 사용**
- 모델을 등록함으로 **모델 관리**



비지니스 로직

Business Logic

Controller

MemberController.java
MemberControllerImpl.java



Service@Transactional

MemberService.java
MemberServiceImpl.java



Repository@CRUD

MemberDAO.java
MemberDAOImpl.java



```
@Controller("memberController")
@RequestMapping(value="/member")
public class MemberControllerImpl extends BaseController implements MemberController{
    @Autowired
    private MemberService memberService;
    @Autowired
    private MemberVO memberVO;

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

        @Repository("memberDAO")
        public class MemberDAOImpl implements MemberDAO{
            @Autowired
            private SqlSession sqlSession;

            <bean id="sqlSession"
                  class="org.mybatis.spring.SqlSessionTemplate">
                <constructor-arg ref="sqlSessionFactory"></constructor-arg>
            </bean >

            <!-- 테이블 -->
            <!-- T_GOODS_DETAIL_IMAGE -->
            <!-- T_SHOPPING_CART -->
            <!-- T_SHOPPING_GOODS -->
            <!-- T_SHOPPING_MEMBER -->
        }
    }
}
```

간단하게 정리하자면, DB로부터 정보를 sqlSession이 받아서 @Repository, @Service, @Controller 순으로 전달합니다.

@Repository는 mybatis.mapper와 연결하여 CRUD 문법을 수행합니다.

@Service는 Repository와 Controller의 중간에서 transactional의 기능을 수행합니다.

@Controller는 중간 제어자 역할을 수행하며 사용자의 요청을 처리 한 후 지정된 View를 반환합니다.

메인페이지

Book ShoppingMall

로그인 | 회원가입 | 고객센터 |

BookTopia

검색

국내외 도서

- IT/인터넷
- 경제/경영
- 대학교재
- 자기계발
- 자연과학/공학
- 역사/인문학

음반

- 가요
- 록
- 클래식
- 뉴에이지
- 재즈
- 기타

N Pay 11월 이벤트

- 11월에도 포인트 더블!! 무조건 2% 적립!
- 간편결제 이용시 최대 4천원 적립!

공지사항

BookTopia

로그인 | 회원가입 | 고객센터 |

모두의 딥러닝

원리를 쉽게 이해하고 나만의 딥러닝 모델을 만들 수 있다.

페이스북 트위터 RSS 피드

최근 본 상품
상품이 없습니다.

베스트셀러

제목	가격
모두의 딥러닝	24,000원
마인크래프트 게임제작	14,000원
판테크 가계부	30,000원
부동산 상식 사전	20,000원

제목	가격
Try! helloworld 자바스크립트	26,000원
직장인을 위한 예술 실무	26,000원
시나공 워드 프로세서 실기	26,000원
컴퓨터 활용능력 2급 실기	26,000원

국지사한입니다 1 26,000원 26,000원 26,000원 26,000원

기능	세부기능
메인페이지	화면 요소
	빠른 메뉴
회원 기능	로그인
	회원 가입
상품 기능	상품 검색
장바구니 기능	장바구니 상품 추가
	장바구니 상품 조회
	장바구니 상품 수정
주문 기능	주문 상품 조회
	주문 상품 구매(payUp)
마이페이지 기능	주문 내역 조회
	회원 정보 수정
	회원 탈퇴

회원 기능

필수입력사항

아이디	<input type="text"/>	<input type="button" value="중복체크"/>			
비밀번호	<input type="password"/>				
이름	<input type="text"/>				
성별	<input type="radio"/> 여성	<input checked="" type="radio"/> 남성			
법정생년월일	2000 <input type="button" value="년"/>	5 <input type="button" value="월"/>	10 <input type="button" value="일"/>	<input checked="" type="radio"/> 양력	<input type="radio"/> 음력
전화번호	없음 <input type="button" value="▼"/>	- <input type="text"/>	- <input type="text"/>		
휴대폰번호	010 <input type="button" value="▼"/>	- <input type="text"/>	- <input type="text"/>		
<input checked="" type="checkbox"/> 소핑몰에서 발송하는 SMS 소식을 수신합니다.					
이메일 (e-mail)	<input type="text"/> @ <input type="text"/> <input type="button" value="직접입력"/>	<input checked="" type="checkbox"/> 소핑몰에서 발송하는 e-mail을 수신합니다.			
주소	<input type="text"/> 우편번호검색 지번 주소: <input type="text"/>				
	도로명 주소: <input type="text"/>				
	나머지 주소: <input type="text"/>				

기능	세부기능
메인페이지	화면 요소
	빠른 메뉴
회원 기능	로그인
	회원 가입
상품 기능	상품 검색
장바구니 기능	장바구니 상품 추가
	장바구니 상품 조회
	장바구니 상품 수정
주문 기능	주문 상품 조회
	주문 상품 구매(payUp)
마이페이지 기능	주문 내역 조회
	회원 정보 수정
	회원 탈퇴

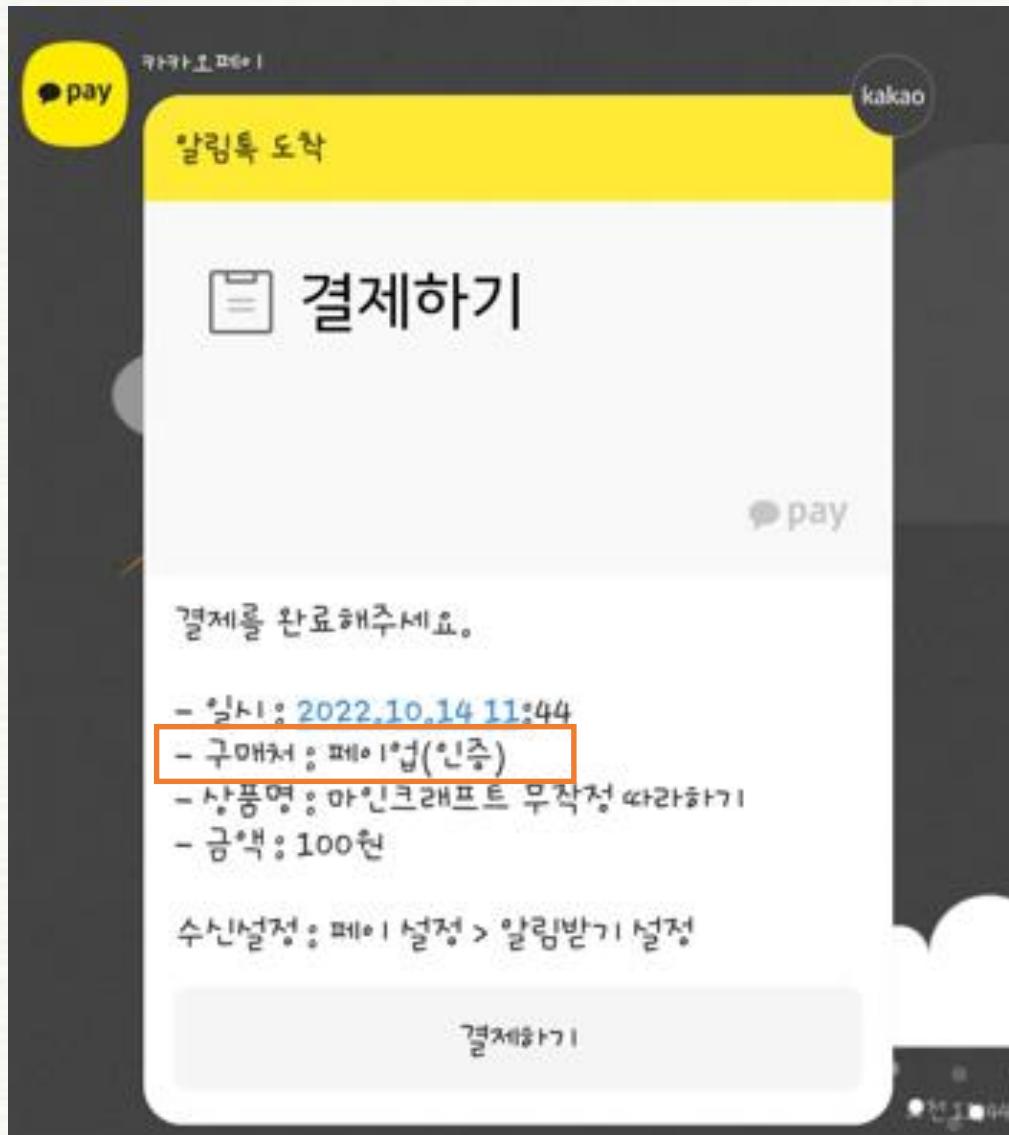
장바구니 기능

로그아웃 | 마이페이지 | 장바구니 | 주문배송 | 고객센터 | 검색

구분	상품명	정가	판매가	수량	합계	주문
<input checked="" type="checkbox"/> 모두의 딥러닝	모두의 딥러닝	24000원	19,440원(10%할인)	1 <input type="button" value="변경"/>	19,440원	<input type="button" value="주문하기"/> <input type="button" value="나중에 주문"/> <input type="button" value="리스트 넣기"/> <input type="button" value="삭제"/>
<input checked="" type="checkbox"/> 판테크 가게부	판테크 가게부	30000원	27,000원(10%할인)	1 <input type="button" value="변경"/>	27,000원	<input type="button" value="주문하기"/> <input type="button" value="나중에 주문"/> <input type="button" value="리스트 넣기"/> <input type="button" value="삭제"/>
<input checked="" type="checkbox"/> 부동산 상식 사전	부동산 상식 사전	20000원	18,000원(10%할인)	1 <input type="button" value="변경"/>	18,000원	<input type="button" value="주문하기"/> <input type="button" value="나중에 주문"/> <input type="button" value="리스트 넣기"/> <input type="button" value="삭제"/>
총 상품수: 3개 총 상품금액: 64,440원 총 배송비: 0원 총 할인 금액: 0원 최종 결제금액: 64,440원						
<input type="button" value="주문하기"/> <input type="button" value="쇼핑계속하기"/>						

기능	세부기능
메인페이지	화면 요소
	빠른 메뉴
회원 기능	로그인
	회원 가입
상품 기능	상품 검색
장바구니 기능	장바구니 상품 추가 장바구니 상품 조회 장바구니 상품 수정
주문 기능	주문 상품 조회 주문 상품 구매(payUp)
마이페이지 기능	주문 내역 조회 회원 정보 수정 회원 탈퇴

주문 기능



기능	세부기능
메인페이지	화면 요소
	빠른 메뉴
회원 기능	로그인
	회원 가입
상품 기능	상품 검색
장바구니 기능	장바구니 상품 추가
	장바구니 상품 조회
	장바구니 상품 수정
주문 기능	주문 상품 조회
	주문 상품 구매(payUp)
마이페이지 기능	주문 내역 조회
	회원 정보 수정
	회원 탈퇴

마이페이지 기능

로그아웃 | 마이페이지 | 장바구니 | 주문배송 | 고객센터 |

검색

내 상세 정보

아이디	hi		
비밀번호	수정하기	
이름	hi		
성별	<input type="radio"/> 여성 <input checked="" type="radio"/> 남성	수정하기	
법정생년월일	2000 년 5 월 10 일	<input checked="" type="radio"/> 양력 <input type="radio"/> 음력	수정하기
전화번호	없음 - <input type="text"/> - <input type="text"/> 010 - <input type="text"/> - <input type="text"/>	수정하기	
휴대폰번호	<input type="checkbox"/> 쇼핑몰에서 발송하는 SMS 소식을 수신합니다.	수정하기	
이메일(e-mail)	j@naver.com,non	직접입력 ▼	수정하기
	<input type="checkbox"/> 쇼핑몰에서 발송하는 e-mail을 수신합니다.		
주소	<input type="button" value="우편번호검색"/> 지번 주소: <input type="text"/> 도로명 주소: <input type="text"/> 수정하기 나머지 주소: <input type="text"/>		

기능	세부기능
메인페이지	화면 요소
	빠른 메뉴
회원 기능	로그인
	회원 가입
상품 기능	상품 검색
장바구니 기능	장바구니 상품 추가 장바구니 상품 조회 장바구니 상품 수정
주문 기능	주문 상품 조회 주문 상품 구매(payUp)
마이페이지 기능	주문 내역 조회 회원 정보 수정 회원 탈퇴

메인 페이지 - 화면요소

header



Tiles

```
<groupId>org.apache.tiles</groupId>
<artifactId>tiles-core</artifactId>
<version>2.2.2</version>
</dependency>
<dependency>
    <groupId>org.apache.tiles</groupId>
    <artifactId>tiles-jsp</artifactId>
    <version>2.2.2</version>
</dependency>
<dependency>
    <groupId>org.apache.tiles</groupId>
    <artifactId>tiles-servlet</artifactId>
    <version>2.2.2</version>
</dependency>
```

CSS

```
1 @font-face {
2     font-family: NanumGothic;
3     src: local("?"), url(.. imgs/nanum.woff) format(woff);
4 }
5 @* {
6     margin: 0px; padding: 0px; vertical-align: middle;
7 }
8 @body {
9     background-position: top; margin: 0px; padding: 0px;
10 }
11 @img {
12     border: currentColor; border-image: none;
13 }
14 @a {
15     color: #rgb(102, 102, 102); text-decoration: none;
16 }
17 @a:hover {
18     color: #rgb(255, 102, 0); text-decoration: none;
19 }
20 @clear {
21     clear: both;
22 }
23 @#outer_wrap {
24     margin: 0px auto; padding: 0px; width: 1100px;
25 }
26 @#wrap {
27     margin: 0px auto; padding: 0px; width: 995px;
28 }
29 @header {
30     margin-top: 20px;
31 }
32 @header #logo {
33     margin-left: 10px; float: left;
```

apache.tiles

```
<dependency>
<id>org.apache.tiles</id>
</dependency>
```

webapp

```
    <!-- resources -->
        <!-- css -->
            basic-jquery-slider.css
            main.css
```

CSS

- header
- aside
- article
- stiky
- footer

-각각의 margin, padding 등
보여지는 요소를 지정

메인 페이지 - 화면요소

WEB-INF / views / common / * .jsp

WEB-INF / views / common / main.jsp

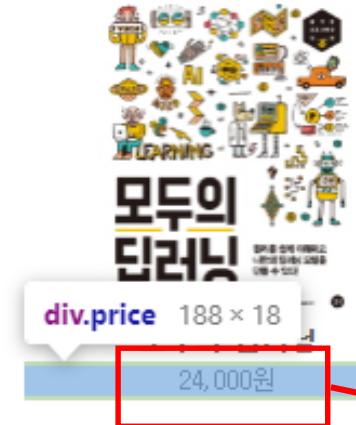
WEB-INF / views / common / tiles_main.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE tiles-definitions PUBLIC
    "-//Apache Software Foundation//DTD Tiles Configuration 2.0//EN"
    "http://tiles.apache.org/dtds/tiles-config_2_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="쇼핑몰 메인페이지" />
        <put-attribute name="body" value="/WEB-INF/views/main/main.jsp" />
    </definition>
</tiles-definitions>
```

tiles로 구성한 baseLayout은 각각의 *.jsp의 화면을 나내냄으로 페이지 전환이 이루어 질 때마다
각각의 layout의 페이지 변환 없이 body(main.jsp)의 고정 된 페이지를 나타내는 것이 가장 큰 장점

메인 페이지 - 화면요소

베스트셀러



goods / service / GoodsServiceImpl.java

```
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("newbook");
    goodsMap.put("newbook", goodsList);

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

```
s_id=354">
    
</a>

<div class="title">모두의 딥러닝</div>
<div class="price"> 24,000원 </div> == $0
...
    </div>
    > <div class="book">...</div>
    > <div class="book">...</div>
```

WEB-INF / views / main / main.jsp

```
<h3>베스트셀러</h3>
<c:forEach var="item" items="${goodsMap.bestseller }">
    <c:set var="goods_count" value="${goods_count+1 }" />
    <div class="book">
        <a href="${contextPath}/goods/goodsDetail.do?goods_id=${i
        ${item.goods_title }</div>
        <div class="price">
            <fmt:formatNumber var="goods_price" type="number"
                value="${item.goods_price}" />
            ${goods_price}원
        </div>
    </div>
```

베스트 셀러 아래의 24,000원이
어떻게 어떤 과정을 거쳐
브라우저에 출력되는지
알아보겠습니다.

메인 페이지 - 화면요소

베스트셀러



div.price 188 × 18
24,000원

goods / service / GoodsServiceImpl.java

```
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("newbook");
    goodsMap.put("newbook", goodsList);

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

```
s_id=354">
    
</a>

<div class="title">모두의 딥러닝</div>
<div class="price"> 24,000원 </div> == $0
</div>
▶ <div class="book">...</div>
    </div>
```

...

WEB-INF / views / main / main.jsp

```
<h3>베스트셀러</h3>
<c:forEach var="item" items="${goodsMap.bestseller }">
    <c:set var="goods_count" value="${goods_count+1 }" />
    <div class="book">
        <a href="${contextPath}/goods/goodsDetail.do?goods_id=${i
        ${item.goods_title }</div>
            <div class="price">
                <fmt:formatNumber var="goods_price" type="number"
                    value="${item.goods_price}" />
                ${goods_price}원
            </div>
        </div>
    </div>
```

- GoodsServiceImpl.java의 Map<String, List<GoodsVO>> 타입의 listGoods() 함수로 goodsMap 객체를 만듭니다.
- 그 객체 안에 "bestseller"를 put합니다.
- main.jsp에서는 goodsMap.bestseller로 받습니다.

메인 페이지 - 화면요소

베스트셀러



div.price 188 × 18
24,000원

goods / service / GoodsServiceImpl.java

```
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("newbook");
    goodsMap.put("newbook", goodsList);

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

```
s_id=354">
    
</a>

<div class="title">모두의 딥러닝</div>
<div class="price"> 24,000원 </div> == $0
</div>
▶ <div class="book">...</div>
    ...
```

WEB-INF / views / main / main.jsp

```
<h3>베스트셀러</h3>
<c:forEach var="item" items="${goodsMap.bestseller }">
    <c:set var="goods_count" value="${goods_count+1 }" />
    <div class="book">
        <a href="${contextPath}/goods/goodsDetail.do?goods_id=${i
        ${item.goods_title }</div>
        <div class="price">
            <fmt:formatNumber var="goods_price" type="number"
                value="${item.goods_price}" /> ${goods_price}원
        </div>
    </div>
```

- Map타입의 goodsMap.bestseller의 정보를 jsp.JSTL태그 기능을 사용하여 items의 값을 var="item"에 넣습니다

- \${item.goods_price}는 GoodsVO에서 goods_price의 값을 받습니다.

메인 페이지 - 화면요소

베스트셀러



div.price 188 × 18

24,000원

```
s_id=354">
    
</a>

    <div class="title">모두의 딥러닝</div>
    <div class="price"> 24,000원 </div> == $0
</div>
    ><div class="book">...</div>
    ><div class="book">...</div>
    ><div class="book">...</div>
    ><div class="book">...</div>
    ><div class="book">...</div>
    ><div class="book">...</div>
```

goods / service / GoodsServiceImpl.java

```
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("newbook");
    goodsMap.put("newbook", goodsList);

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

WEB-INF / views / main / main.jsp

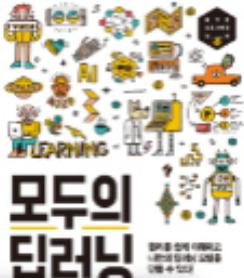
```
<h3>베스트셀러</h3>
<c:forEach var="item" items="${goodsMap.bestseller }">
    <c:set var="goods_count" value="${goods_count+1 }" />
    <div class="book">
        <a href="${contextPath}/goods/goodsDetail.do?goods_id=${i
        ${item.goods_title }</div>
        <div class="price">
            <fmt:formatNumber var="goods_price" type="number"
                value="${item.goods_price}" /> ${goods_price}원
        </div>
    </div>
```

기존에는

<%=session.getAttribute("goods_price") %>로
표현했지만 Expression Language는
jsp 2.0에서 추가된 스크립트 언어로
EL을 이용하여 세션에 있는
var="goods_price"를
\$(goods_price)로 브라우저에 전송합니다.

메인 페이지 - 화면요소

베스트셀러



div.price 188 × 18

24,000원

goods / service / GoodsServiceImpl.java

```
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("newbook");
    goodsMap.put("newbook", goodsList);

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

```
s_id=354">
    
</a>

    <div class="title">모두의 딥러닝</div>
    <div class="price"> 24,000원 </div> == $0
    ...
    <div class="book">...</div>
    <div class="book">...</div>
    <div class="book">...</div>
    <div class="book">...</div>
    <div class="book">...</div>
    <div class="book">...</div>
```

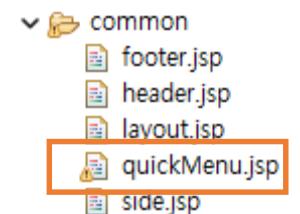
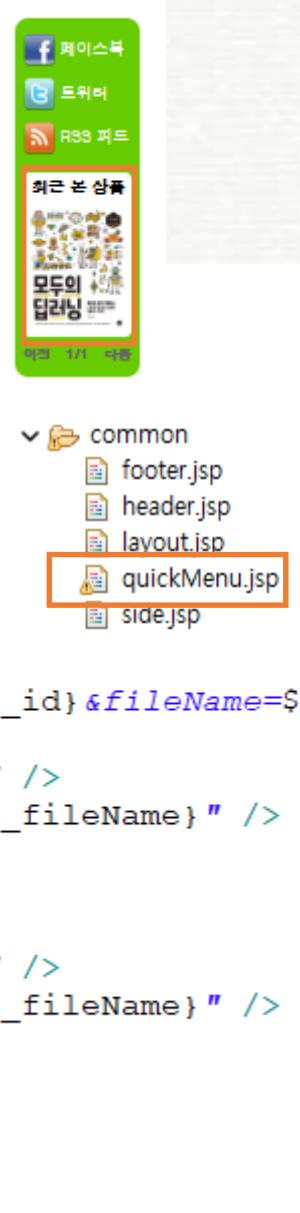
WEB-INF / views / main / main.jsp

```
<h3>베스트셀러</h3>
<c:forEach var="item" items="${goodsMap.bestseller }">
    <c:set var="goods_count" value="${goods_count+1 }" />
    <div class="book">
        <a href='${contextPath}/goods/goodsDetail.do?goods_id=${i
        ${item.goods_title }</div>
            <div class="price">
                <fmt:formatNumber var="goods_price" type="number"
                    value="${item.goods_price}" /> ${goods_price}원
            </div>
        </div>
```

메인 페이지 - 빠른 메뉴

WEB-INF / views / common / quickMenu.jsp

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<div id="sticky" >
<c:choose>
<c:when test="${empty quickGoodsList } ">
    <strong>상품이 없습니다.</strong>
</c:when>
<c:otherwise>
<form name="frm_sticky" >
<c:forEach var="item" items="${quickGoodsList } " varStatus="itemNum">
<c:choose>
    <c:when test="${itemNum.count==1 } ">
        <a href="javascript:goodsDetail();">
            
            <input type="hidden" name="h_goods_id" value="${item.goods_id} " />
            <input type="hidden" name="h_goods_fileName" value="${item.goods_fileName} " />
        </a>
        <br>
    </c:when>
    <c:otherwise>
        <input type="hidden" name="h_goods_id" value="${item.goods_id} " />
        <input type="hidden" name="h_goods_fileName" value="${item.goods_fileName} " />
    </c:otherwise>
    </c:choose>
</c:forEach>
</c:otherwise>
</c:choose>
```



quickMenu.jsp

sticky

- jsp/jstl/core <TagLibrary>
- <c : choose/>
<c : when , otherwise/>
jstl에서 제공하는 if, else문
- forEach
jstl에서 제공하는 for문
- varStatus
상태변수라고 하며
주로 사용되는 것은
current : 현재 해당 번호
count : 1부터 시작
- img_sticky
thumbnails를 사용하여
var="item"에 차례로 저장
- hidden
상품을 클릭할 때마다
상품의 정보를 숨기는 역할

메인 페이지 - 빠른 메뉴

WEB-INF / views / common / quickMenu.jsp

```
var array_index=0;
var SERVER_URL="${contextPath}/thumbnails.do";
function fn_show_next_goods(){
    var img_sticky=document.getElementById("img_sticky");
    var cur_goods_num=document.getElementById("cur_goods_num");
    var _h_goods_id=document.frm_sticky.h_goods_id;
    var _h_goods_fileName=document.frm_sticky.h_goods_fileName;
    if(array_index < _h_goods_id.length-1)
        array_index++;

    var goods_id=_h_goods_id[array_index].value;
    var fileName=_h_goods_fileName[array_index].value;
    img_sticky.src=SERVER_URL+"?goods_id="+goods_id+"&fileName="+fileName;
    cur_goods_num.innerHTML=array_index+1;
}

<c:choose>
<c:when test="${empty quickGoodsList }">
    <h5>    0/0    </h5>
</c:when>
<c:otherwise>
    <h5><a href='javascript:fn_show_previous_goods();'> 이전 </a> ${quickGoodsListNum}   <a href='javascript:fn_show_next_goods();'> 다음 </a> </h5>
    <span id='cur_goods_num'>1</span>/${quickGoodsListNum}
</c:otherwise>
</c:choose>
```



주황

- 다음 버튼을 누르게 되면 fn_show_next_goods() 호출

검정

- 다른 상품을 검색할 때마다 cur_goods_num의 인덱스를 1씩 증가

초록

- session에 있는 img_sticky의 값을 받아와 'url + ?' 매개변수 goods_id를 thumbnails를 사용하여 나타나게 합니다

회원 기능 - 로그인 기능

WEB-INF / views / member / loginForm.jsp

```
<body>
    <H3>회원 로그인 창</H3>
    <DIV id="detail_table">
        <form action="${contextPath}/member/login.do" method="post">
            <TABLE>
                <TBODY>
                    <TR class="dot_line">
                        <TD class="fixed_join">아이디</TD>
                        <TD><input name="member_id" type="text" size="20" /></TD>
                    </TR>
                    <TR class="solid_line">
                        <TD class="fixed_join">비밀번호</TD>
                        <TD><input name="member_pw" type="password" size="20" /></TD>
                    </TR>
                </TBODY>
            </TABLE>
            <br><br>
            <INPUT type="submit" value="로그인"/>
            <INPUT type="button" value="초기화"/>

            <br><br>
            <a href="#">아이디 찾기</a> | 
            <a href="#">비밀번호 찾기</a> | 
            <a href="${contextPath}/member/memberForm.do">회원가입</a> | 
            <a href="#">고객 센터</a>
        </form>
    </body>
```

loginForm.jsp에서 구현한
회원 로그인 창에서
로그인 버튼을 눌렀을 때,
MemberController.java에서 login함수를
MemberControllerImpl.java로 실행합니다.

member / controller / MemberController.java

```
public interface MemberController {
    public ModelAndView login(@RequestParam Map<String, String> loginMap, HttpServlet
    public ModelAndView logout(HttpServletRequest request, HttpServletResponse resp
    public ResponseEntity addMember(@ModelAttribute("member") MemberVO member,
        HttpServletRequest request, HttpServletResponse response) throws Excepti
    public ResponseEntity overlapped(@RequestParam("id") String id, HttpServletReq
}
```

회원 기능 - 로그인 기능

member / controller / MemberControllerImpl.java

```
@Override  
@RequestMapping(value="/login.do", method = RequestMethod.POST)  
public ModelAndView login(@RequestParam Map<String, String> loginMap,  
                           HttpServletRequest request, HttpServletResponse response,  
                           ModelAndView mav) {  
    memberVO=memberService.login(loginMap);  
    if(memberVO!= null && memberVO.getMember_id()!=null){  
        HttpSession session=request.getSession();  
        session.setAttribute("isLogOn", true);  
        session.setAttribute("memberInfo", memberVO);  
  
        String action=(String)session.getAttribute("action");  
        if(action!=null && action.equals("/order/orderEachGoods.do")){  
            mav.setViewName("forward:"+action);  
        }else{  
            mav.setViewName("redirect:/main/main.do");  
        }  
  
    }else{  
        String message="아이디나 비밀번호가 틀립니다. 다시 로그인해주세요";  
        mav.addObject("message", message);  
        mav.setViewName("/member/loginForm");  
    }  
    return mav;  
}
```

member / dao / MemberDAOImpl.java

```
@Override  
public MemberVO login(Map<String, String> loginMap) throws DataAccessException{  
    MemberVO member=(MemberVO)sqlSession.selectOne("mapper.member.login", loginMap);  
    return member;
```

resources / mybatis / member.xml

```
<select id="login" resultType="memberVO" parameterType="java.util.Map" >  
    <! [CDATA[  
        select * from t_shopping_member  
        where member_id=#{member_id}  
        and member_pw=#{member_pw}  
    ]>
```

- ID와 비밀번호를 **Map**타입으로 **loginMap**에 저장합니다.
- 저장된 정보를 **member.xml**의 **Query문**으로 조회를 합니다.
- **memberVO.getMember_id()**가 null값이 아닌 경우 **session**을 가져옵니다.
- **session**의 해당 id값을 **isLogOn : session**에 활성화 중으로 만듭니다.
- 해당 **memberVO**값의 **memberInfo**를 **session**에 머무르게 합니다.

회원 기능 - 회원가입 기능 - 중복체크

로그인 | 회원가입 | 고객센터 |

검색

필수입력사항

아이디 중복체크

비밀번호

이름

성별 여성 남성

법정생년월일 년 월 일 양력 음력

전화번호 - -

휴대폰번호 - -

쇼핑몰에서 발송하는 SMS 소식을 수신합니다.

이메일 (e-mail) @ 직접입력

쇼핑몰에서 발송하는 e-mail을 수신합니다.

주소
도로명 주소:
나머지 주소:

회원가입 다시입력

MemberControllerImpl.java

```
@Override  
@RequestMapping(value="/overlapped.do" ,method = RequestMethod.POST)  
public ResponseEntity overlapped(@RequestParam("id") String id,HttpSer  
    ResponseEntity resEntity = null;  
    String result = memberService.overlapped(id);  
    resEntity =new ResponseEntity(result, HttpStatus.OK);  
    return resEntity;
```

MemberDAOImpl.java

```
@Override  
public String selectOverlappedID(String id) throws DataAccessException {  
    String result = sqlSession.selectOne("mapper.member.selectOverlappedID",id);  
    return result;
```

member.xml

```
<select id="selectOverlappedID" parameterType="String" resultType="String">  
    select decode(count(*),1,'true',0,'false')  
    from t_shopping_member  
    where member_id = #{id}
```

select decode (count(*) , 1 , 'true' ,0 , 'false') from t_shopping_member where member_id = # { id }
-from t_shopping_member where member_id = # { id }
입력된 id 값이 스키마에 있는지 조회
-select decode (count(*) , 1 , 'true' ,0 , 'false')
count(*) 함수는 테이블의 레코드 수를 반환해서 해당 1 값이 있으면 true를
반환하고 0 없으면 false를 반환

회원 기능 - 회원가입 기능 - 중복체크

memberForm.jsp

필수입력사항

아이디	addMember	중복체크
비밀번호	
이름	진명국	
성별	<input type="radio"/> 여성 <input checked="" type="radio"/> 남성	
법정생년월일	1989년 12월 5일	
전화번호	없음	-
	010 - 9201	- 1205
휴대폰번호	<input checked="" type="checkbox"/> 쇼핑몰에서 발송하는 SMS 소	
이메일 (e-mail)	jinmkook	@naver.com
	<input checked="" type="checkbox"/> 쇼핑몰에서 발송하는 e-mail을	
	13112	우편번호검색
	지번 주소: 경기 성남시 수정구 복정로96번길 26	
주소	도로명 주소: 경기 성남시 수정구 복정동 669-3	
	나머지 주소:	
<input type="button" value="회원가입"/> <input type="button" value="다시입력"/>		

```
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) {
        }
    });
}
```

\$.ajax({

- Key : "Value", 값을 비동기 정보로 처리합니다.
 - url값은 memberController의 overlapped.do로 전송됩니다.
 - data : { id:_id } member.xml에서 받은 decode값이 true이면 해당 id는 있는 것으로 그 값이 false 일 때 session에 member_id값을 저장합니다.
-)

회원가입

- 정보가 입력이 된 상태로 회원가입 버튼을 누르면 session에 입력된 값을 Controller에게 전송하고 mybatis를 통해 DB에 적재합니다.

상품 기능 - 상품 검색

GoodsControllerImpl.java

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



```
<select id="selectGoodsBySearchWord" resultMap="goodsResult" parameterType="string" >
  <! [CDATA[
    select g.* , d.fileName from t_shopping_goods g, t_goods_detail_image d
    where g.goods_id=d.goods_id
    and d.filetype='main_image'
    and g.goods_title like '%'|| #{searchWord} || '%'
    order by g.goods_creDate desc
  ]]>
</select>
```

goods.xml

like '%' || #{searchWord} || '%'

- like %W => boW, coW, followW
- like W% => World, War, Warm
- like %W% => 문자열 중간에 W가 포함 된 모든 문자열을 포함

장바구니 기능

MEMBER

COLUMN_NAME	DATA_TYPE
1 MEMBER ID	VARCHAR2 (20 BYTE)
2 MEMBER PW	VARCHAR2 (30 BYTE)
3 MEMBER NAME	VARCHAR2 (50 BYTE)
4 MEMBER GENDER	VARCHAR2 (10 BYTE)
5 TEL1	VARCHAR2 (20 BYTE)
6 TEL2	VARCHAR2 (20 BYTE)
7 TEL3	VARCHAR2 (20 BYTE)
8 HP1	VARCHAR2 (20 BYTE)
9 HP2	VARCHAR2 (20 BYTE)
10 HP3	VARCHAR2 (20 BYTE)
11 SMSSTS YN	VARCHAR2 (20 BYTE)
12 EMAIL1	VARCHAR2 (20 BYTE)
13 EMAIL2	VARCHAR2 (20 BYTE)
14 EMAILSTS YN	VARCHAR2 (20 BYTE)
15 ZIPCODE	VARCHAR2 (20 BYTE)
16 ROADADDRESS	VARCHAR2 (500 B...)
17 JIBUNADDRESS	VARCHAR2 (500 B...)
18 NAMUJIADDRESS	VARCHAR2 (500 B...)
19 MEMBER BIRTH Y	VARCHAR2 (20 BYTE)
20 MEMBER BIRTH M	VARCHAR2 (20 BYTE)
21 MEMBER BIRTH D	VARCHAR2 (20 BYTE)
22 MEMBER BIRTH GN	VARCHAR2 (20 BYTE)
23 JOINDATE	DATE
24 DEL YN	VARCHAR2 (20 BYTE)

MEMBER-CART-GOODS Entity Relationship Diagram

CART

COLUMN_NAME	DATA_TYPE
1 CART ID	NUMBER (10, 0)
2 GOODS ID	NUMBER (20, 0)
3 MEMBER ID	VARCHAR2 (20 BYTE)
4 DEL YN	VARCHAR2 (20 BYTE)
5 CREDITATE	DATE
6 CART GOODS QTY	NUMBER (4, 0)

-CART ID는
GOODS ID와
MEMBER ID의
값을 갖으면서
ERD구조를
갖게 됩니다.

GOODS

COLUMN_NAME	DATA_TYPE
1 GOODS ID	NUMBER (20, 0)
2 GOODS SORT	VARCHAR2 (50 BYTE)
3 GOODS TITLE	VARCHAR2 (100 B...)
4 GOODS WRITER	VARCHAR2 (50 BYTE)
5 GOODS PUBLISHER	VARCHAR2 (50 BYTE)
6 GOODS PRICE	NUMBER (10, 0)
7 GOODS SALES PRICE	NUMBER (10, 0)
8 GOODS POINT	NUMBER (10, 0)
9 GOODS PUBLISHED...	DATE
10 GOODS TOTAL PAGE	NUMBER (5, 0)
11 GOODS ISBN	VARCHAR2 (50 BYTE)
12 GOODS DELIVERY ...	NUMBER (10, 0)
13 GOODS DELIVERY ...	DATE
14 GOODS STATUS	VARCHAR2 (50 BYTE)
15 GOODS INTRO	VARCHAR2 (2000 ...)
16 GOODS WRITER INTRO	VARCHAR2 (2000 ...)
17 GOODS PUBLISHER...	VARCHAR2 (2000 ...)
18 GOODS RECOMMEND...	VARCHAR2 (2000 ...)
19 GOODS CONTENTS ...	CLOB
20 GOODS CREDITATE	DATE

장바구니 기능 - 상품 추가

정가 32,000원
판매가 28,800원(10%할인)
포인트적립 1000P(10%적립)
포인트 추가적립 만원이상 구매시 1,000P, 5만원이상 구매시 2,000P 추가적립 편의점 배송 이용시 300P 추가적립
발행일 2018-08-01
페이지 수 688쪽
ISBN 9791160505238
배송료 무료
배송안내 [당일배송] 당일배송 서비스 시작!
[휴일배송] 휴일에도 배송받는 Bookshop
도착예정일 지금 주문 시 내일 도착 예정
수량 1

구매하기 장바구니 위시리스트

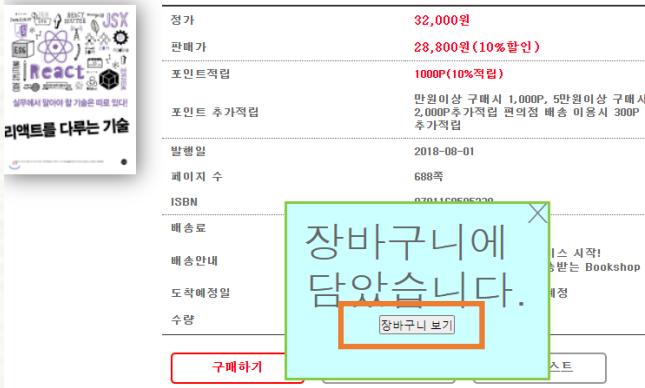
cart.xml

```
<insert id="insertGoodsInCart" parameterType="cartVO">
  <![CDATA[
    insert into t_shopping_cart(cart_id,
      goods_id,
      member_id)
    values(#{cart_id},
      #{goods_id},
      #{member_id})
  ]]>
</insert>
```

CART	
COLUMN_NAME	DATA_TYPE
1 CART_ID	NUMBER(10, 0)
2 GOODS_ID	NUMBER(20, 0)
3 MEMBER_ID	VARCHAR2(20 BYTE)
4 DEL_YN	VARCHAR2(20 BYTE)
5 CREDITATE	DATE
6 CART_GOODS_QTY	NUMBER(4, 0)

- 장바구니 기능의 상품 추가는 상품이 있는지 없는지를 먼저 조회하고 상품이 없으면 cart 스키마에 적재합니다.
- insertGoodsInCart는 insert into t_shopping_cart를 실행 (해당 key) value (해당 values)

장바구니 기능 - 상품 조회



구분	상품명	정가	판매가	수량	합계	주문
<input checked="" type="checkbox"/> 모두의 딥러닝	모두의 딥러닝	24000원	19,440원(10% 할인)	<input type="button" value="1"/> 변경	19,440원	<input type="button" value="주문하기"/> 나중에 주문 리스트 넣기 삭제
<input checked="" type="checkbox"/> 판테크 가게부	판테크 가게부	30000원	27,000원(10% 할인)	<input type="button" value="1"/> 변경	27,000원	<input type="button" value="주문하기"/> 나중에 주문 리스트 넣기 삭제
<input checked="" type="checkbox"/> 부동산 상식 사전	부동산 상식 사전	20000원	18,000원(10% 할인)	<input type="button" value="1"/> 변경	18,000원	<input type="button" value="주문하기"/> 나중에 주문 리스트 넣기 삭제
<input checked="" type="checkbox"/> 리액트를 다루는 기술	리액트를 다루는 기술	32000원	28,800원(10% 할인)	<input type="button" value="1"/> 변경	28,800원	<input type="button" value="주문하기"/> 나중에 주문 리스트 넣기 삭제

myCartList.jsp

```
<form name="frm_order_all_cart">
    <c:forEach var="item" items="${myGoodsList }" varStatus="cnt">
        <c:set var="cart_goods_qty" value="${myCartList[cnt.count-1].cart_goods_qty }" />
        <c:set var="cart_id" value="${myCartList[cnt.count-1].cart_id }" />
        <td><input type="checkbox" name="checked_goods" checked value="${item.goods_id }" onClick="calcGoodsPrice(${item.goods_sales_price },this)"></td>
        <td class="goods_image">
            <a href="${contextPath}/goods/goodsDetail.do?goods_id=${item.goods_id }">
                
            </a>
        </td>
```

myCartList.jsp

- 조회된 상품의 목록을 출력합니다.

- 주문하기, 삭제 등 각 항목에 를 걸어서 클릭하면 해당하는 곳으로 이동하도록 합니다.

- 왼쪽에 있는 썸네일을 클릭했을 때 goodsDetail.do?goods_id='매개변수'로 이동하는 링크입니다.

장바구니 기능 - 상품 수정

The screenshot shows the project structure in a Java IDE:

- cart
- controller
- CartController.java
- CartControllerImpl.java

The browser screenshot displays a shopping cart page with the following details:

제품	상품명	정가	판매가	수량	합계	주문
<input checked="" type="checkbox"/>	모두의 딥러닝	24000원	19,440원(10% 할인)	<input type="text" value="2"/> 변경	19,440원	<button>주문하기</button> 나중에 주문 리스트 넣기 삭제
<input checked="" type="checkbox"/>	마인크래프트 무작정 따라하기	14000원	10,800원(10% 할인)	<input type="text" value="1"/> 변경	10,800원	<button>주문하기</button> 나중에 주문 리스트 넣기 삭제

At the bottom, there are summary statistics:

총 상품수	총 상품금액	총 배송비	총 할인 금액	최종 결제 금액
2개	30,240원	[+]	0원	[-] 0원 = 30,240원

Buttons at the bottom include **주문하기** and **쇼핑 계속하기**.

CartControllerImpl.java

```
@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)

    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";
    }
}
```

modifyCartQty함수는 수정될 때 상품 정보와 회원 정보를 수정하고 주문 수량을 수정하여 수정된 CartService의 결과에 저장하여 반환하고 상품 수량을 반환합니다.

주문 기능

MEMBER

COLUMN_NAME	DATA_TYPE
1 MEMBER_ID	VARCHAR2 (20 BYTE)
2 MEMBER_PW	VARCHAR2 (30 BYTE)
3 MEMBER_NAME	VARCHAR2 (50 BYTE)
4 MEMBER_GENDER	VARCHAR2 (10 BYTE)
5 TEL1	VARCHAR2 (20 BYTE)
6 TEL2	VARCHAR2 (20 BYTE)
7 TEL3	VARCHAR2 (20 BYTE)
8 HP1	VARCHAR2 (20 BYTE)
9 HP2	VARCHAR2 (20 BYTE)
10 HP3	VARCHAR2 (20 BYTE)
11 SMSSTS_YN	VARCHAR2 (20 BYTE)
12 EMAIL1	VARCHAR2 (20 BYTE)
13 EMAIL2	VARCHAR2 (20 BYTE)
14 EMAILSTS_YN	VARCHAR2 (20 BYTE)
15 ZIPCODE	VARCHAR2 (20 BYTE)
16 ROADADDRESS	VARCHAR2 (500 B...)
17 JIBUNADDRESS	VARCHAR2 (500 B...)
18 NAMUJIADDRESS	VARCHAR2 (500 B...)
19 MEMBER_BIRTH_Y	VARCHAR2 (20 BYTE)
20 MEMBER_BIRTH_M	VARCHAR2 (20 BYTE)
21 MEMBER_BIRTH_D	VARCHAR2 (20 BYTE)
22 MEMBER_BIRTH_GN	VARCHAR2 (20 BYTE)
23 JOINDATE	DATE
24 DEL_YN	VARCHAR2 (20 BYTE)

ORDER

COLUMN_NAME	DATA_TYPE
1 ORDER_SEQ_NUM	NUMBER (20, 0)
2 ORDER_ID	NUMBER (20, 0)
3 MEMBER_ID	VARCHAR2 (20 BYTE)
4 GOODS_ID	NUMBER (20, 0)

order_seq_num

- 주문자가 개별로 주문 상품을 구분하는 번호
- seq_num은 각각이 갖는 1개 당의 고유번호이고

order_id

- 주문자들의 주문을 구분하는 번호
- order_id는 그 책이 갖는 번호입니다

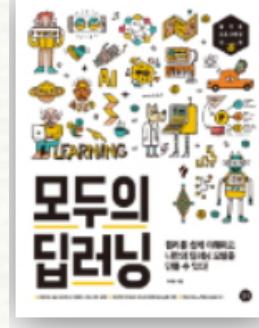
GOODS

COLUMN_NAME	DATA_TYPE
1 GOODS_ID	NUMBER (20, 0)
2 GOODS_SORT	VARCHAR2 (50 BYTE)
3 GOODS_TITLE	VARCHAR2 (100 B...)
4 GOODS_WRITER	VARCHAR2 (50 BYTE)
5 GOODS_PUBLISHER	VARCHAR2 (50 BYTE)
6 GOODS_PRICE	NUMBER (10, 0)
7 GOODS_SALES_PRICE	NUMBER (10, 0)
8 GOODS_POINT	NUMBER (10, 0)
9 GOODS_PUBLISHED...	DATE
10 GOODS_TOTAL_PAGE	NUMBER (5, 0)
11 GOODS_ISBN	VARCHAR2 (50 BYTE)
12 GOODS_DELIVERY ...	NUMBER (10, 0)
13 GOODS_DELIVERY ...	DATE
14 GOODS_STATUS	VARCHAR2 (50 BYTE)
15 GOODS_INTRO	VARCHAR2 (2000 ...)
16 GOODS_WRITER_INTRO	VARCHAR2 (2000 ...)
17 GOODS_PUBLISHER...	VARCHAR2 (2000 ...)
18 GOODS_RECOMMEND...	VARCHAR2 (2000 ...)
19 GOODS_CONTENTS ...	CLOB
20 GOODS_CREDITATE	DATE

주문 기능 - 주문 상품 조회

모두의 딥러닝

조태호 저 | 길벗



- 정가
- 판매가
- 포인트적립
- 포인트 추가적립
- 발행일
- 페이지 수
- ISBN
- 배송료
- 배송안내
- 도착예정일
- 수량

구매하기

- views
 - > admin
 - > cart
 - > common
 - > goods
 - > main
 - > member
 - > mypage
 - > order
 - orderGoodsForm.jsp**
 - orderReslutKakao.jsp
 - orderResult.jsp
 - orderResultFail.jsp

```
<tr class="dot_line">
    <td class="fixed_join">받으실 분</td>
    <td><input id="receiver_name" name="receiver_name" type="text" size="40" value="${orderer.member_name}" /> <input type="hidden" id="h_orderer_name" name="h_orderer_name" value="${orderer.member_name }" /> <input type="hidden" id="h_receiver_name" name="h_receiver_name" value="${orderer.member_name }" /></td>
```

1. 주문 확인

주문상품명	수량	주문금액	배송비	예상적립금	주문금액합계
모두의 딥러닝	1개	21600원(10% 할인)	0원	1500원	21600원

2. 배송지 정보

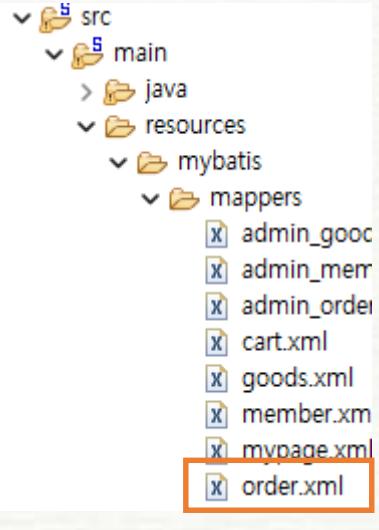
배송방법	<input checked="" type="radio"/> 일반택배	<input type="radio"/> 편의점택배	<input type="radio"/> 해외배송
배송지 선택	<input checked="" type="radio"/> 기본배송지	<input type="radio"/> 새로입력	<input type="radio"/> 최근배송지
받으실 분	진명국		
휴대폰번호	010	-	9201 - 1205
유선전화(선택)	<input type="text"/> - <input type="text"/> - <input type="text"/>		
주소	<p>13112 우편번호검색 지번 주소: 경기 성남시 수정구 복정로96번길 26 (복정동)</p> <p>도로명 주소: 경기 성남시 수정구 복정동 669-3</p> <p>나머지 주소: <input type="text"/></p>		
배송 메시지	택배 기사님께 전달할 메시지를 남겨주세요.		
선물 포장	<input type="radio"/> 예 <input checked="" type="radio"/> 아니요		

'구매하기' 버튼을 누르면
orderGoodsForm.jsp
페이지로 이동합니다.

orderGoddsForm.jsp에서
session값을 반환합니다.

주문 기능 - 주문 상품 조회

order.xml



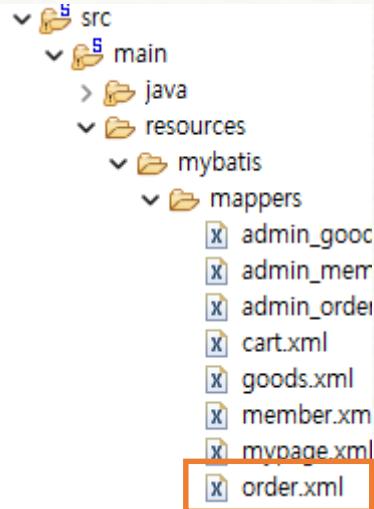
```
<insert id="insertNewOrder" parameterType="orderVO" >
    <selectKey resultType="int" keyProperty="order_seq_num" order="BEFORE">
        select order_seq_num.nextval from dual
    </selectKey>
    <![CDATA[
        insert into t_shopping_order(order_seq_num,
            order_id,
            member_id,
            goods_id,
            goods_title,
            goods_fileName,
            order_goods_qty,
            pay_orderer_hp_num)
        values(${order_seq_num},
            ${order_id},
            ${member_id},
            ${goods_id},
            ${goods_title},
            ...
            )
        ${card_com_name},
        ${card_pay_month},
        ${orderer_hp},
        ${pay_orderer_hp_num})
    ]]>
</insert>

<select id="selectOrderID" resultType="int" >
    <![CDATA[
        select seq_order_id.nextval from dual
    ]]>
</select>
```

- Query문을 insert할 때 mybatis에서 제공하는 **selectKey**태그를 이용하여 일련번호를 가져오는 SQL문을 따로 하지 않아도 됩니다.
- 이와 같이 시퀀스에서 주문상품일련번호를 가져와 주문 테이블에 추가할 수 있습니다.
- Insert문으로 추가된 내용으로 selectOrderID로 주문아이디를 가져옵니다.
from dual은 dummy table의 약자입니다.

주문 기능 - 주문 상품 조회

order.xml



```

<insert id="insertNewOrder" parameterType="orderVO" >
    <selectKey resultType="int" keyProperty="order_seq_num" order="BEFORE">
        select order_seq_num.nextval from dual
    </selectKey>
    <![CDATA[
        insert into t_shopping_order(order_seq_num,
            order_id,
            member_id,
            goods_id,
            goods_title,
            goods_fileName,
            order_goods_qty,
            pay_orderer_hp_num)
        values(${order_seq_num},
            ${order_id},
            ${member_id},
            ${goods_id},
            ${goods_title},
            ${goods_fileName},
            ${order_goods_qty},
            ${pay_orderer_hp_num})
    ]]>
</insert>

<select id="selectOrderID" resultType="int" >
    <![CDATA[
        select seq_order_id.nextval from dual
    ]]>
</select>

```

1. 주문 확인

주문상품명	수량	주문금액	배송비	예상적립금	주문금액합계
모두의 딥러닝	1개	21600원(10% 할인)	0원	1500원	21600원

2. 배송지 정보

주문고객	이름 <input type="text" value="진명국"/>
핸드폰	<input type="text" value="010-9201-1205"/>
이메일	<input type="text" value="jinkook@naver.com"/>
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	<input type="text" value="진명국"/>
휴대폰번호	010 <input type="text" value="9201"/> - <input type="text" value="1205"/>
유선전화(선택)	<input type="text"/> - <input type="text"/> - <input type="text"/>
우편번호검색	<input type="text" value="13112"/> <input type="button" value="지번 주소"/>
주소	도로명 주소: <input type="text" value="경기 성남시 수정구 복정로96번길 26 (복정동)"/>
나머지 주소:	<input type="text"/>
배송 메시지	택배 기사님께 전달할 메시지를 남겨주세요. <input type="text"/>
선물 포장	<input type="radio"/> 예 <input checked="" type="radio"/> 아니요

주문 기능 - 주문 상품 구매

Payment



2차 PG사 – payup에서
강의한 내용으로
신용 카드 구매와
카카오페이(간편결제)가 있는데
카카오페이만
다루도록 하겠습니다.

The screenshot shows the payup payment interface. At the top, there's a navigation bar with 'HOME', 'MENU', and user information ('himedia [테스트회사] * (과세 사업자)님 감사합니다', '59분 34초', 'LOG OUT'). Below the menu, there are three sections: '오늘 거래내역' (Today's Transaction History) showing 500 원, '이번달 거래내역' (This Month's Transaction History) showing 45,800 원, and '이번달 정산내역' (This Month's Settlement History) showing 0 원. Each section has tables for '신용카드' (Credit Card), '간편결제' (Convenience Payment), and '가상계좌' (Virtual Account). At the bottom, there's a link '증가 거래내역' (Increase Transaction History).

카드번호 :

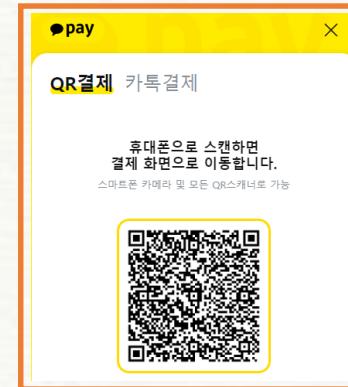
유효기간 월 :

유효기간 년 :

생년월일 :

비밀번호2자리 :

최종 결제하기



| 승인 : 6 건 / 57,100 원

| 부분취소 : 0 건 / 0 원

| 전체취소 : 0 건 / 0 원

| 승인실패 : 10 건

<input type="checkbox"/> 승인일시	취소일시	상품명	고객명	승인번호	카드종류	휴대폰	간편결제	금액	할부개월	상태	매출전표
<input type="checkbox"/> 2022.10.13 14:43:57		자바와 파이썬	진명국	47061611	KB국민카드 (5365-10*****7356)			10,000원	일시불	승인성공	발급

주문 기능 - 주문 상품 구매

pom.xml

```
<!-- https://mvnrepository.com/artifact/com.squareup.okhttp3/okhttp -->
<dependency>
    <groupId>com.squareup.okhttp3</groupId>
    <artifactId>okhttp</artifactId>
    <version>3.10.0</version>
</dependency>

<!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-databind -->
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
    <version>2.13.3</version>
</dependency>
```

TestService.java

```
public String encrypt(String text) throws NoSuchAlgorithmException {
    MessageDigest md = MessageDigest.getInstance("SHA-256");
    md.update(text.getBytes());

    return bytesToHex(md.digest());
}

private String bytesToHex(byte[] bytes) {
    StringBuilder builder = new StringBuilder();
    for (byte b : bytes) {
        builder.append(String.format("%02x", b));
    }
    return builder.toString();
}
```

okhttp3, jackson-databind를 lib에 추가
- okhttp3는 **REST API, HTTP 통신을 간편하게 사용**
- jackson은 json형태의 데이터 구조를 처리

SHA-256 알고리즘

- 256비트로 구성, 블록체인, 암호화에 특화
- 2^{256} 의 경우수를 만들어 내는 암호화

주문 기능 – 주문 상품 구매

```
@RequestMapping(value= "/test/kakaoOrder.do")
public Map<String, Object> kakaoOrder(@RequestParam Map<String, String> dateMap) throws Exception{

    // {amount=25000, itemName=직장인을 위한 엑셀 실무, userName=진명국}
    System.out.println(dateMap.toString());

    String id = "himedia"; //발급된 계정
    String base = "https://api.testpayup.co.kr";
    String path = "/ep/api/kakao/" + id + "/order";

    String url = base + path;

    String signature = "";

    //파라미터로 사용할 맵
    Map<String, String> map = new HashMap<String, String>();
    map.put("orderNumber", "TEST_12345"); //주문데이터 생성해야지 생기는 키값 (생성을 안해서 임의값)
    map.put("amount", dateMap.get("amount")); //화면에서 받아온 값
    map.put("amount", "100"); //화면에서 받아온 값 (테스트를 위해서 100원 고정)
    map.put("itemName", dateMap.get("itemName")); //화면에서 받아온 값
    map.put("userName", dateMap.get("userName")); //화면에서 받아온 값
    map.put("userName", "테스트중"); //(테스트를 위해서 고정)
    map.put("returnUrl", "naver.com"); //PC에서는 안중요함
    map.put("timestamp", "1000");
    map.put("userAgent", "WP");

    signature = apiService.encrypt(id + "+" + map.get("orderNumber") + "+" + map.get("amount") + "|ac805b30517f4fd08e3e80490e559f8e|" + map.get("timestamp"));

    map.put("signature", signature);

    Map<String, Object> resultMap = new HashMap<String, Object>();
    resultMap = apiService.restApi(map, url);

    // {good_mny=25000, site_cd=A8QOB, Ret_URL=naver.com, affiliaterCode=0005, buyr_name=진명국, ordr_idxx=20221014100929KK0905, good_name=직장인을 위한 엑셀 실무,
    System.out.println(resultMap.toString());

    System.out.println("응답코드 = " + resultMap.get("responseCode"));

    return resultMap;
}
```

PG사에 접근하기 위해서는 계정을 발급 받아야 합니다.

signature의 encrypt(암호화한 값을) 반환해서 카카오 페이에 전달합니다.

주문 기능 - 주문 상품 구매

4. 결제정보

신용카드 제휴 신용카드 실시간 계좌이체 무통장 입금
 휴대폰 결제 카카오페이지(간편결제) 페이나우(간편결제)
직접입금
카드 선택:
할부 기간:

orderGoodsForm.jsp

```
/* 이 함수를 실행하여 카카오결제창을 호출 합니다*/
function jsf_pay() {
try {
var form = document.order_info;
KCP_Pay_Execute(form);
} catch{
/* IE에서 결제 정상종료시 throw로 스크립트 종료 */
}
}
</script>
<script type="text/javascript"
src="https://pay.kcp.co.kr/plugin/payplus_web.jsp"></script>
```



카카오 페이에서 제공하는 jsf_pay()를 불러 옵니다.

주문 기능 – 주문 상품 구매

```
function fn_show_order_detail() {
    //결제를 무엇을 선택했는지 체크
    var payType = $('input[name="pay_method"]:checked').val();
    if(payType == '카카오페이지(간편결제') {
        //화면에서 가져갈 데이터
        var amount = $("#h_each_goods_price").val(); //결제금액
        var itemName = $("#h_goods_title").val(); //상품명
        var userName = $("#h_orderer_name").val(); //구매자

        //카카오페이지 선택
        //여기안에서
        //차바에서 주문api요청을 한 후 데이터 받아오기
        //비동기 ajax 사용 예정
        // alert("카카오페이지 선택"); //알림메세지
        $.ajax({
            type : "post",
            async : false, //false인 경우 동기식으로 처리한다.
            url : "${contextPath}/test/kakaoOrder.do",
            data : {
                "amount":amount
                , "itemName":itemName
                , "userName":userName
            },
            success : function(data, textStatus) {
                console.log(data);
                //컨트롤러에서 받은 데이터를 화면에 넣기

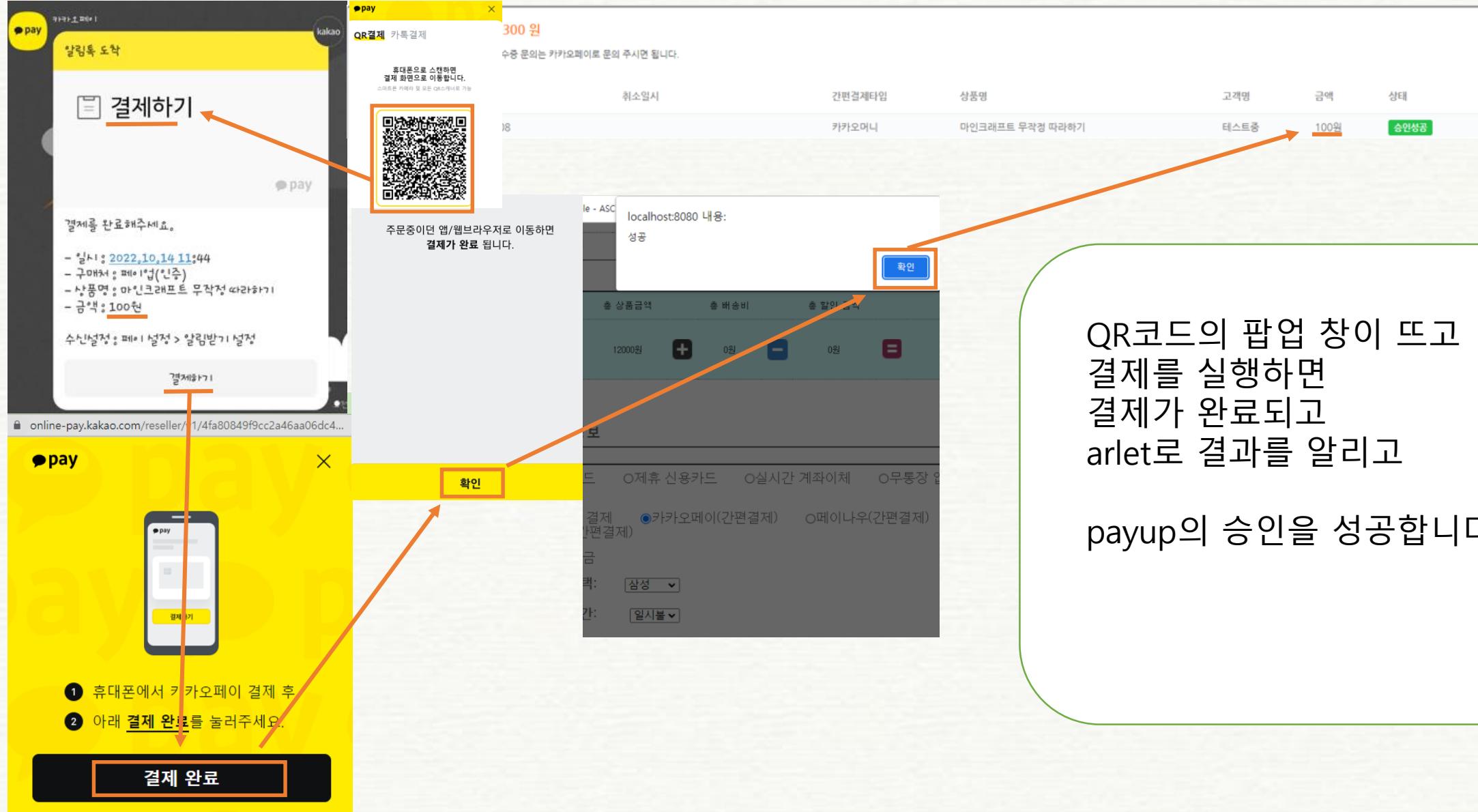
                if(data.responseCode == '0000'){ //주문을 성공했을 때
                    주문요청 후 응답 받은 데이터를 form에 넣음
                    $('input[name="ordr_idxx"]').val(data.ordr_idxx);
                    $('input[name="good_name"]').val(data.good_name);
                    $('input[name="good_mny"]').val(data.good_mny);
                    $('input[name="buyr_name"]').val(data.buyr_name);
                    $('input[name="site_cd"]').val(data.site_cd);

                    jsf__pay() 함수를 호출
                    jsf__pay();
                }
            }
        });
    }
}
```

checked.val(); = 카카오페이지(간편결제)일 때

- **\$.ajax({key : value 형식의 비동기 값을 브라우저에서 추출해 값을 jsf__pay();함수를 호출합니다.});**
- **ajax를 사용하는 이유는 동기값의 DB와 연동할 수 없기 때문에 session의 값들을 조회해야 하기 때문입니다.**
- **값은 카카오 페이에 전송되고 결제 창을 불러와 결제를 승인 받습니다.**

주문 기능 - 주문 상품 구매



마이 페이지 기능 - 주문내역 조회

로그아웃 마이페이지 장바구니 주문배송 고객센터

검색

mypage.xml

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

MyPageControllerImpl.java

```
@Override
@RequestMapping(value="/myPageMain.do", method = RequestMethod.GET)
public ModelAndView myPageMain(@RequestParam(required = false, value="message")
    HttpServletRequest request, HttpServletResponse response) throws
    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;
}
```

최근주문내역				
주문번호	주문일자	주문상품	주문상태	주문취소
402	2022-10-06 20:20:53.0	모두의 딥러닝/1개 부동산 상식 사전/1개 직장인을 위한 엑셀 실무/1개	배송준비중	주문취소

검정

- 마이 페이지 버튼을 누르면 session의 `side_menu`를 `my_page`로 지정해 줍니다.

주황

- `/myPageMain.do`를 호출하면 위의 그림과 같이 회원 아이디를 이용하여 주문 상품을 조회합니다.

초록

- `order by time desc` 내림차순을 사용하여 최근에 주문한 상품부터 조회하게 합니다.

마이 페이지 기능 - 회원 정보 수정

내 상세 정보

아이디	addMember
비밀번호
이름	진명국
성별	<input type="radio"/> 여성 <input checked="" type="radio"/> 남성
법정생년월일	1989 년 12 월 5 일
전화번호	없음 - <input type="text"/> - <input type="text"/>
휴대폰번호	010 - 9201 - 1205
□ 쇼핑몰에서 발송하는 SMS 소식을 수신합니다.	
이메일 (e-mail)	jinmook @ naver.com.non <input type="button" value="직접입력"/>
□ 쇼핑몰에서 발송하는 e-mail을 수신합니다.	

myDetailInfo.jsp

```
<td class="fixed_join">비밀번호</td>
<td>
    <input name="member_pw" type="password" size="20" value="${memberInfo.member_pw }" />
</td>
<td>
    <input type="button" value="수정하기" onClick="fn_modify_member_info('member_pw')" />
</td>

function fn_modify_member_info(attribute){
    var value;
    var frm_mod_member=document.frm_mod_member;
    if(attribute=='member_pw'){
        value=frm_mod_member.member_pw.value;
    }else if(attribute=='member_gender'){
        var member_gender=frm_mod_member.member_gender;
        for(var i=0; member_gender.length;i++){
            if(member_gender[i].checked){
                value=member_gender[i].value;
                break;
            }
        }
    }

    ajax({
        type : "post",
        async : false,
        url : "${contextPath}/mypage/modifyMyInfo.do",
        data : {
            attribute:attribute,
            value:value,
        },
        success : function(data, textStatus) {
            if(data.trim()=='mod_success'){
                alert("회원 정보를 수정했습니다.");
            }else if(data.trim()=='failed'){
                alert("다시 시도해 주세요.");
            }
        }
    });
}
```

MyPageControllerImpl.java

```
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;
```

- '수정하기' 버튼의 누르면 함수는 attribute속성 값을 if, else if 문으로 해당 name 값을 ajax로 보내고 ajax는 Key, Value 값으로 Controller에 전송합니다.

- Controller에서는 session의 memberInfo를 삭제하고 setAttribute로 등록하고 MemberVO에 전달합니다.

마이 페이지 기능 - 회원 탈퇴

주문내역

- 주문내역/배송 조회
- 반품/교환 신청 및 조회
- 취소 주문 내역
- 세금 계산서

정보내역

- 회원정보관리
- 나의 주소록
- 개인정보 동의내역
- 회원탈퇴

회원 탈퇴 기능을 구현해 보겠습니다.

1. jsp를 만들어 회원 탈퇴를 눌렀을 때 해당 페이지로 이동
2. SQL delete문 확인 -> member.xml 작성
3. DAO -> Service -> Controller 작성
4. jsp와 Controller를 연결
5. 성공 시 마무리

마이 페이지 기능 - 회원 탈퇴 - jsp생성

The screenshot shows the project structure and code for a member deletion feature:

- File Structure:**
 - views/
 - admin
 - cart
 - common/
 - footer.jsp
 - header.jsp
 - layout.jsp
 - quickMenu.jsp
 - side.jsp
 - Ctrl + H (Search for "myDetailInfo")
 - tiles/
 - tiles_mypage.xml (2 matches)
 - 18: <definition name="/mypage/myDetailInfo" extends="baseLayout">
 - 20: <put-attribute name="body" value="/WEB-INF/views/mypage/myDetailInfo.jsp" />
 - webapp/
 - WEB-INF/
 - views/
 - common/
 - side.jsp
 - mypage/
 - deleteInfo.jsp

side.jsp (Content):

```
<li>
    <h3>정보내역</h3>
    <ul>
        <li><a href="${contextPath}/mypage/myDetailInfo.do">회원정보관리</a></li>
        <li><a href="#">나의 주소록</a></li>
        <li><a href="#">개인정보 등의내역</a></li>
        <li><a href="#">회원탈퇴</a></li>
    </ul>
</li>
```

deleteInfo.jsp (Content):

```
localhost:8080 내용:
회원 아이디와 비밀번호를 확인하세요. 탈퇴하기를 누르면 더 이상 서비스를 이용하실 수 없습니다. 그래도 탈퇴하시겠습니까??

아아디 : hi
탈퇴하기
```

- 회원 정보 관리의 myDetailInfo.jsp의 경로를 'Ctrl + H'로 추적하여 'deleteInfo.jsp'를 만들고 테스트합니다.
- 회원의 내용을 표시하는 myDetailInfo.jsp의 필요한 부분만 복사해서 붙여넣습니다.
- 간단한 회원 정보와 탈퇴하기 버튼을 만들고 alert창을 띄워서 신중한 선택을하게 합니다.

마이 페이지 기능 - 회원 탈퇴 - mapper/member.xml

The screenshot shows the Oracle SQL Developer interface. It consists of two main panes: '질의 작성기' (Query Builder) and '질의 결과' (Query Results).
In the top pane, the query is:select * from t_shopping_member;
delete from t_shopping_member where member_id='hello';The second line is highlighted with an orange box.
In the bottom pane, the results of the query are shown in a table:| MEMBER_ID | MEMBER_PW | MEMBER_NAME | MEMBER_GE |
| --- | --- | --- | --- |
| 1 hello | 1212 | hello | 101 |

In the middle section, another '질의 작성기' pane shows the same delete query:

```
select * from t_shopping_member;  
delete from t_shopping_member where member_id='hello';
```

The second line is highlighted with an orange box.
At the bottom, a message box displays the result of the operation:

작업이 완료되었습니다.(0,039초)

1 행 이 (가) 삭제되었습니다.

- delete 문을 작성하고 삭제가 되었는지 확인합니다.
- 실행이 되면 mybatis/meppers/member.xml에 delete문을 작성합니다.

```
<delete id="deleteMember" parameterType="String">  
    <![CDATA[  
        delete from t_shopping_member  
        where member_id=#{member_id};  
    ]]>  
</delete>
```

마이 페이지 기능 - 회원 탈퇴 - DAO-> Ser -> Con

DAO

```
@Override  
public String deleteMember(String member_id) throws DataAccessException {  
  
    String result = sqlSession.delete("mapper.member.deleteMember", member_id);  
    return result;  
}  
  
Type mismatch: cannot convert from int to String
```

1 quick fix available:
[Change type of 'result' to 'int'](#)

UPDATE or DELETE queries can return void or int. If it is an int, the value is
the number of rows affected by this query.

Service

```
@Override  
public int removeMember(String member_id) throws DataAccessException {  
    return memberDAO.deleteMember(member_id);  
}
```

Controller

```
@Override  
public ModelAndView removeMember(String member_id,  
        HttpServletRequest request, HttpServletResponse response)  
throws Exception {  
  
    return null;  
}
```

- DAO를 작성 할 때 에러가 났습니다.
- 검색 해보니 Update와 Deleted의 return타입은 void와 int만 할 수 있고 만약 int라면 값은 Query의 행의 영향을 받는다고 합니다.
- 행 전체를 받기 위해 int를 사용한다면, 해당 member_id가 속해있는 행 전체를 지워야한다고 생각하고 int로 바꿔 줬습니다.
- Controller를 수정해 보겠습니다.

마이 페이지 기능 - 회원 탈퇴 - DAO-> Ser -> Con

Controller

```
@Override  
@RequestMapping(value="/removeMember.do" ,method = RequestMethod.GET)  
public ModelAndView removeMember(String member_id,  
        HttpServletRequest request, HttpServletResponse response)  
        throws Exception {  
  
    request.setCharacterEncoding("utf-8");  
    memberService.removeMember(member_id);  
  
    ModelAndView mav = new ModelAndView();  
    mav.setViewName("redirect:/main/main.do");  
    return mav;  
}
```

```
@Override  
public ModelAndView removeMember(String member_id,  
        HttpServletRequest request, HttpServletResponse response)  
        throws Exception {  
  
    return null;  
}
```

- RequestMapping 은 **/removeMember.do**를 사용합니다.
- method는 받아온 정보를 처리하기 때문에 **GET방식을 사용합니다.**
- **memberService.removeMember**를 사용합니다.
- ModelAndView로 클라이언트가 회원 탈퇴 버튼을 눌렀을때 **main화면으로 보내줍니다.**

마이 페이지 기능 - 회원 탈퇴 - (jsp - > Controller)

The screenshot shows a Java development environment with the following components:

- Top Bar:** Shows tabs for "ASCII Table - ASCII...", "Selenium", and "점프 투 파일...".
- JSP Page Content:** Displays a table with columns: MEMBER_ID, MEMBER_PW, and MEMBER_NAME. Two rows are shown:
 - Row 1: MEMBER_ID = 1, MEMBER_PW = 1212, MEMBER_NAME = hi
 - Row 2: MEMBER_ID = 2, MEMBER_PW = 1212, MEMBER_NAME = 진명국
- Form Fields:** A form with fields "아이디 : hi" and "비밀 번호". Below the form is a button labeled "탈퇴하기".
- Console Output:** Shows a stack trace starting with "Cannot invoke <com.bookshop01.member.vo.MemberVO.getMember_id()> because <this.memberVO> is null".
- Code Editor:** Shows the JSP file code (MyPageControllerImpl.java) with annotations 44 through 49. Annotation 46 contains the link to the delete page.

- 에러의 내용은 VO의 getMember_id를 불러올 수 없다.
VO값이 없기 때문이라는 메세지 였습니다.
- VO값을 어디서 못 가져왔는지 탐색을 시작했습니다.
- member_id 'hi' 를 지우기 위해 포기하지 않고
끝까지 보고 또 본 결과

마이 페이지 기능 - 회원 탈퇴 - (jsp - > Controller)

```
" value="${memberInfo.member_id}" disabled/>  
  
size="20" value="${memberInfo.member_pw}" disabled/>  
  
.do?member_id=${member_id}  
서 감사합니다.' ); ">탈퇴하기</a>
```



The screenshot shows the Book Shopping Mall website's member login page. At the top, there is a navigation bar with links like '로그인' (Login), '회원가입' (Sign Up), '장바구니' (Cart), '주문배송' (Delivery), and '고객센터' (Customer Service). Below the navigation, there is a search bar with the placeholder '검색' (Search) and a button labeled '검색' (Search). The main content area displays a banner for 'Try! helloworld 파이썬' (Try! helloworld Python) and a grid of book thumbnails. On the right side, there is a '워크시트' (Worksheet) window showing a database query and its results.

```
워크시트 질의 작성기  
select * from t_shopping_member;  
  
스크립트 출력 x 질의 결과 x  
SQL | 인출된 모든 행: 3(0,005초)  
MEMBER_ID MEMBER_PW MEMBER  
1 addMember 1212 진명국
```

- deleteInfo.jsp 의 **value="\${memberInfo.member_id}"** 가 눈에 들어왔고 \${ member_id } 를 \${ memberInfo.member_id }로 수정했습니다.
본 PPT 40장에서 쓰인 내용임을 알았습니다.
해당 memberVO값의 **memberInfo**를 session에 머무르게 합니다.
- 결과 메인페이지로 redirect 되었고 DB의 결과도 삭제가 되었습니다.
- 하지만 로그아웃은 되지 않았고 해결하기 위해 session에 남아있는 정보를 제거하기위해 isLogin을 넣어 주었습니다.

마이 페이지 기능 - 회원 탈퇴 - (jsp - > Controller)

```
@Override  
@RequestMapping(value="/removeMember.do" ,method = RequestMethod.GET)  
public ModelAndView removeMember(String member_id,  
        HttpServletRequest request, HttpServletResponse response)  
        throws Exception {  
  
    request.setCharacterEncoding("utf-8");  
    memberService.removeMember(member_id);  
  
    HttpSession session=request.getSession();  
    session.setAttribute("isLogOn", false);  
    session.removeAttribute("memberInfo");  
  
    ModelAndView mav = new ModelAndView();  
    mav.setViewName("redirect:/main/main.do");  
    return mav;  
}
```



isLogOn을 false로 바꿔주고
session에 남아있는 memberInfo의
정보를 제거하기위해
session.removeAttribute("memberInfo")로
정상 작동하게 되었습니다.

이상입니다.

마치며

지금까지 “스프링으로 구현한 쇼핑몰 웹 개발”을 봄하신 분들께 감사드리며
PPT를 마치겠습니다.

감사합니다.