JAVA 프레임워크 SPRING을 이용한 게시판

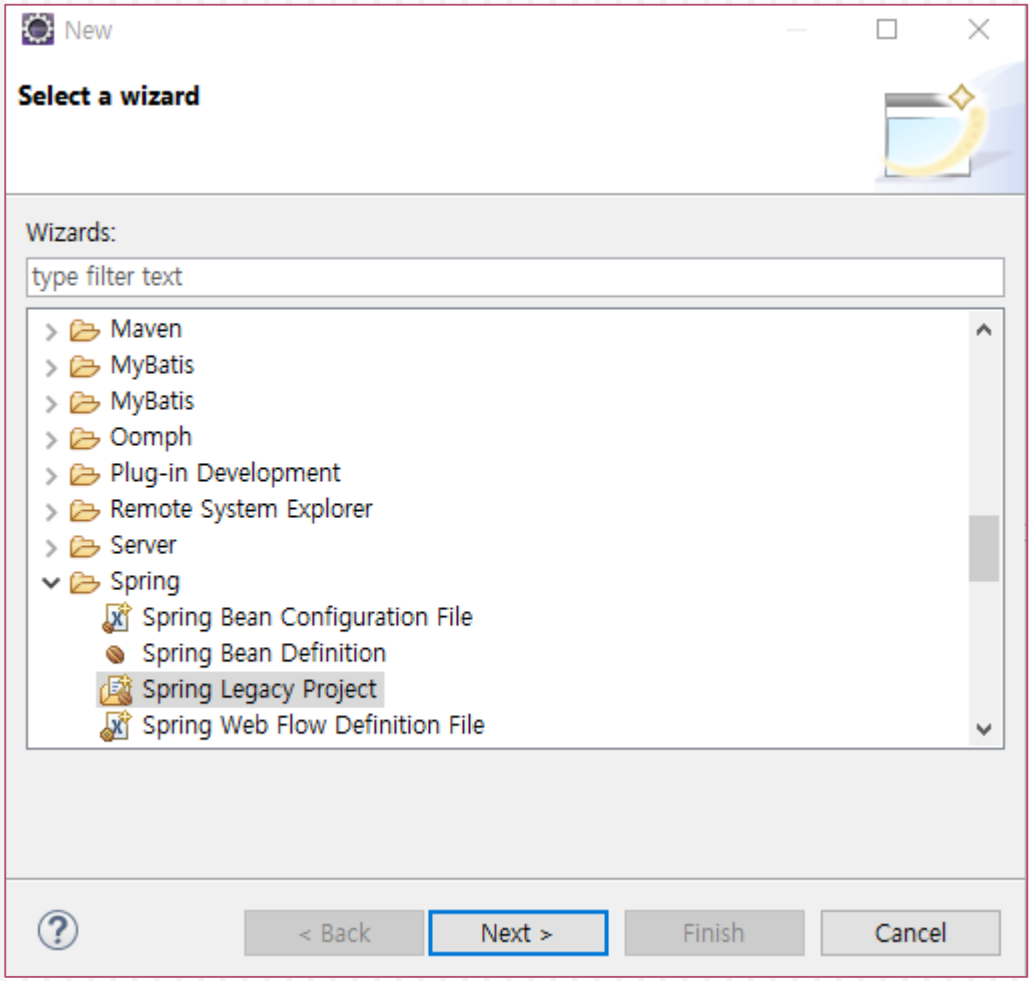
장안대학교
인터넷정보통신과

게시판 프로젝트 진행 순서

- 프로젝트 생성
- Pom.xml 구성
- Web.xml 구성
- Root-context dataSource 작성
- DB 테이블 구성
- DTO 객체 작성 ➔ 테이블과 일치
- Mapper 작성 : xxxMapper.java ➔ xxxMapper.xml
- Root-context SqlSessionFactory 등록
- xxxService.java
- xxxConroller.java
- xxx.jsp

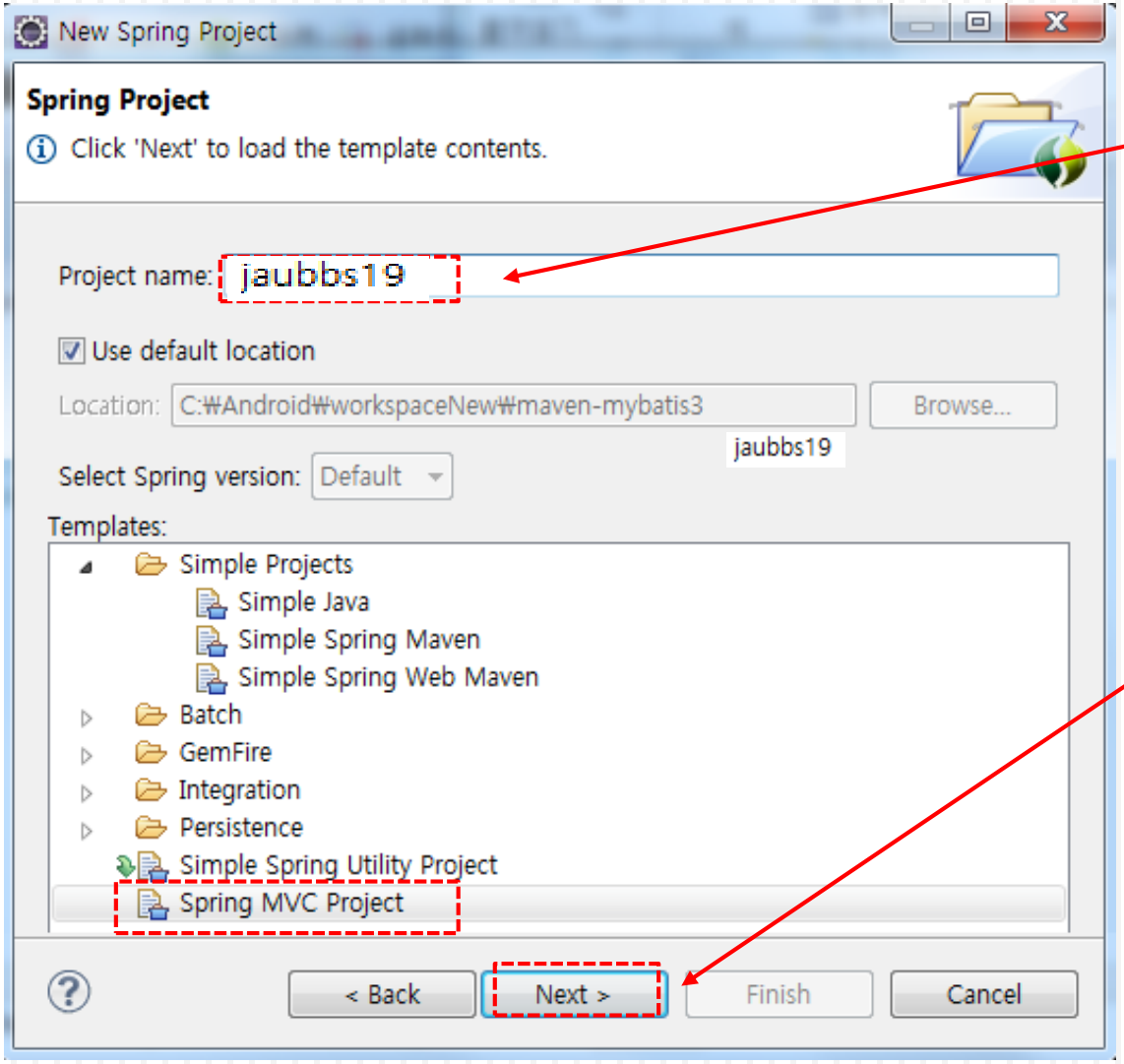
Maven project 구성 (Maven Mybatis)

- File > new> other > Spring > Spring Legacy project > next >



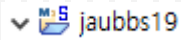
Maven project 구성

- Spring MVC project > next >

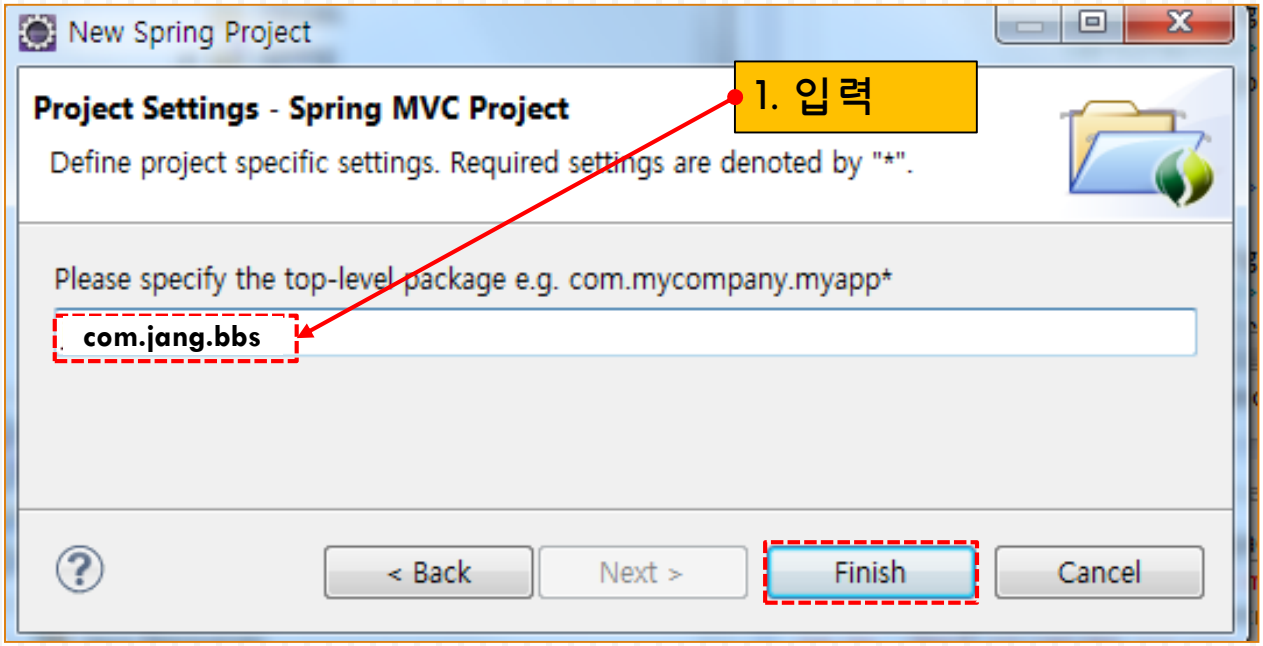


1. 입력

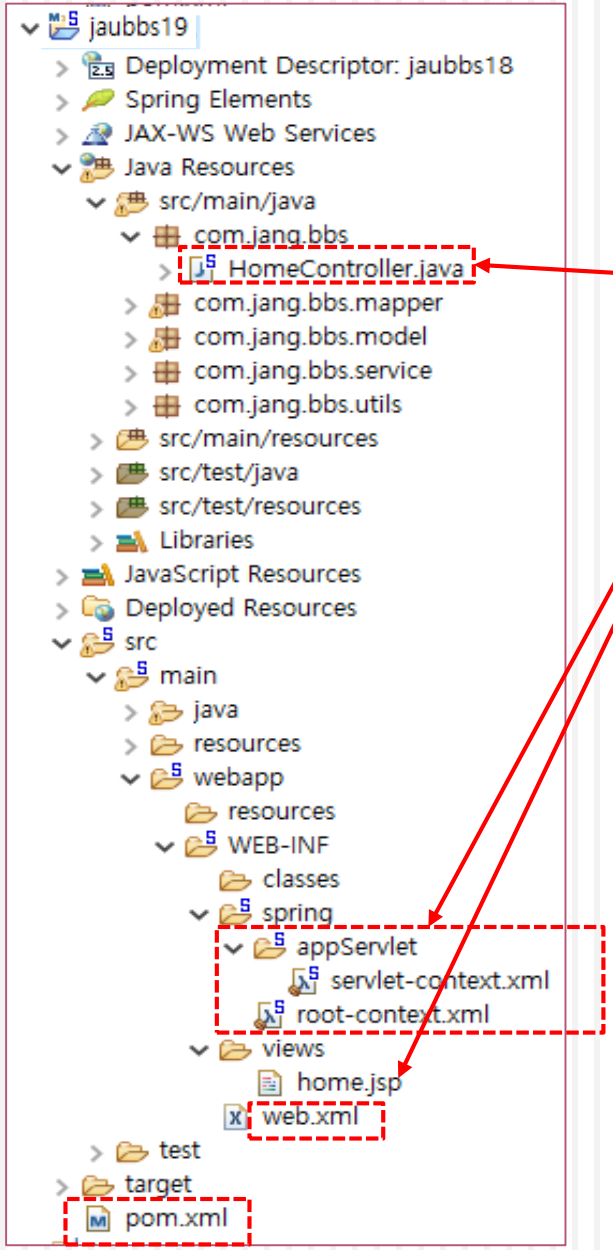
2. 클릭



Maven project 구성



Maven project 구성



1. 기본 구성화일 생성

POM 구성

```
<!-- Spring JDBC Support -->
<dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-jdbc</artifactId>
      <version>${org.springframework-
      version}</version>
</dependency>
<dependency>
    <groupId>com.oracle</groupId>
    <artifactId>ojdbc6</artifactId>
    <version>11.2.0.3</version>
    <scope>compile</scope>
</dependency>
<dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-dbcp2</artifactId>
    <version>2.1.1</version>
</dependency>
<dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-pool2</artifactId>
    <version>2.4.2</version>
</dependency>
<!-- Mybatis Support  -->
<dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis</artifactId>
    <version>3.5.0</version>
</dependency>
<dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis-spring</artifactId>
    <version>2.0.1</version>
</dependency>
```

```
<repositories>
    <repository>
        <id>codelds</id>
        <url>https://code.lds.org/nexus/content/groups/main-repo</url>
    </repository>
</repositories>

<properties>
    <java-version>12</java-version>
    <org.springframework-version>5.0.8.RELEASE</org.springframework-version>
    <org.aspectj-version>1.6.10</org.aspectj-version>
    <org.slf4j-version>1.6.6</org.slf4j-version>
</properties>
```

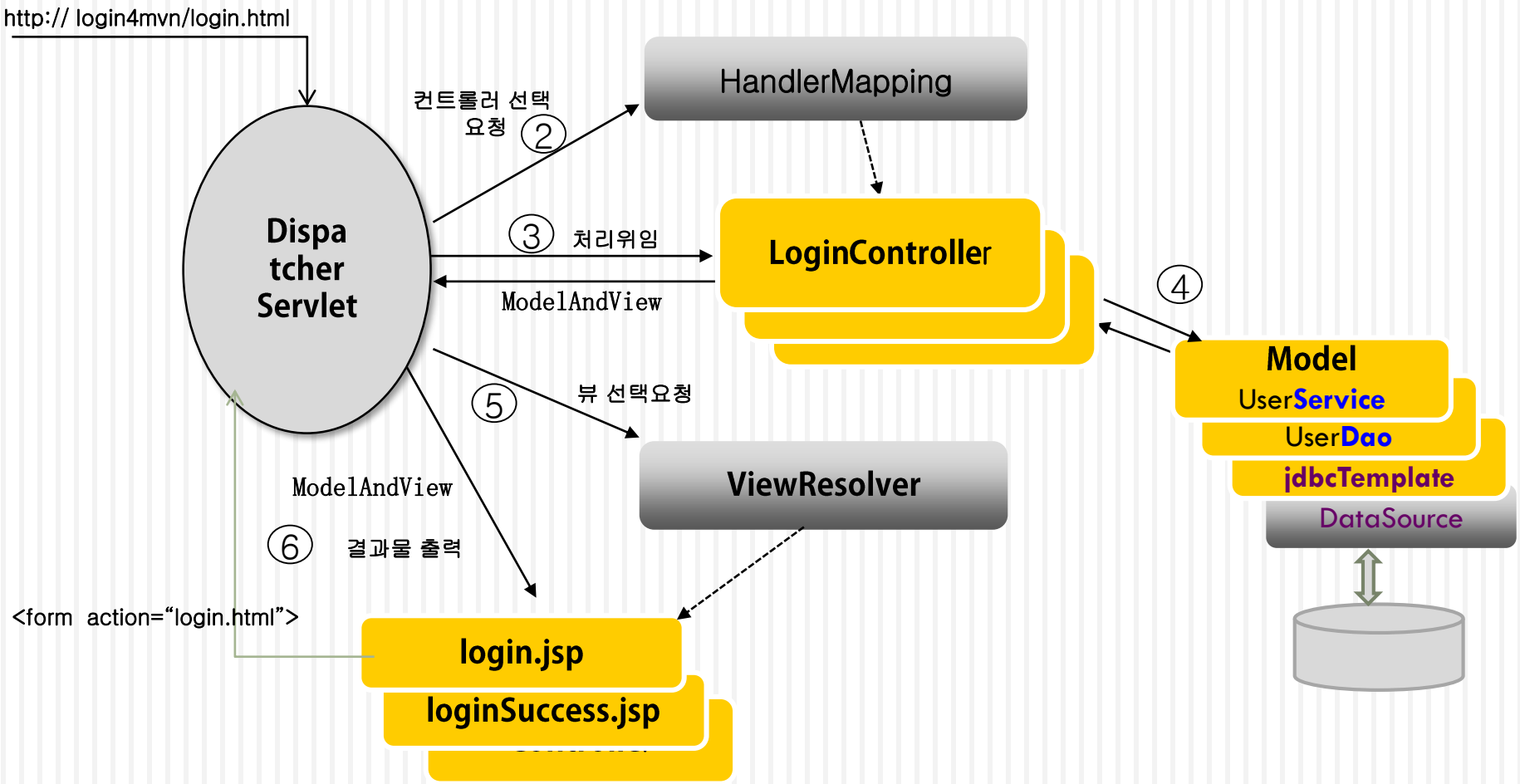
POM 구성

```
<!-- Hibernate Validator -->
<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-validator</artifactId>
    <version>6.0.12.Final</version>
</dependency>
<!-- JSON  Support -->
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-core</artifactId>
    <version>2.9.7</version>
</dependency>
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
    <version>2.9.7</version>
</dependency>
<!-- Apache Commons FileUpload -->
<dependency>
    <groupId>commons-fileupload</groupId>
    <artifactId>commons-fileupload</artifactId>
    <version>1.3.3</version>
</dependency>
<!-- Apache Commons IO -->
<dependency>
    <groupId>commons-io</groupId>
    <artifactId>commons-io</artifactId>
    <version>2.6</version>
</dependency>
<dependency>
  <groupId>javax.xml.bind</groupId>
  <artifactId>jaxb-api</artifactId>
  <version>2.3.0</version>
</dependency>
```

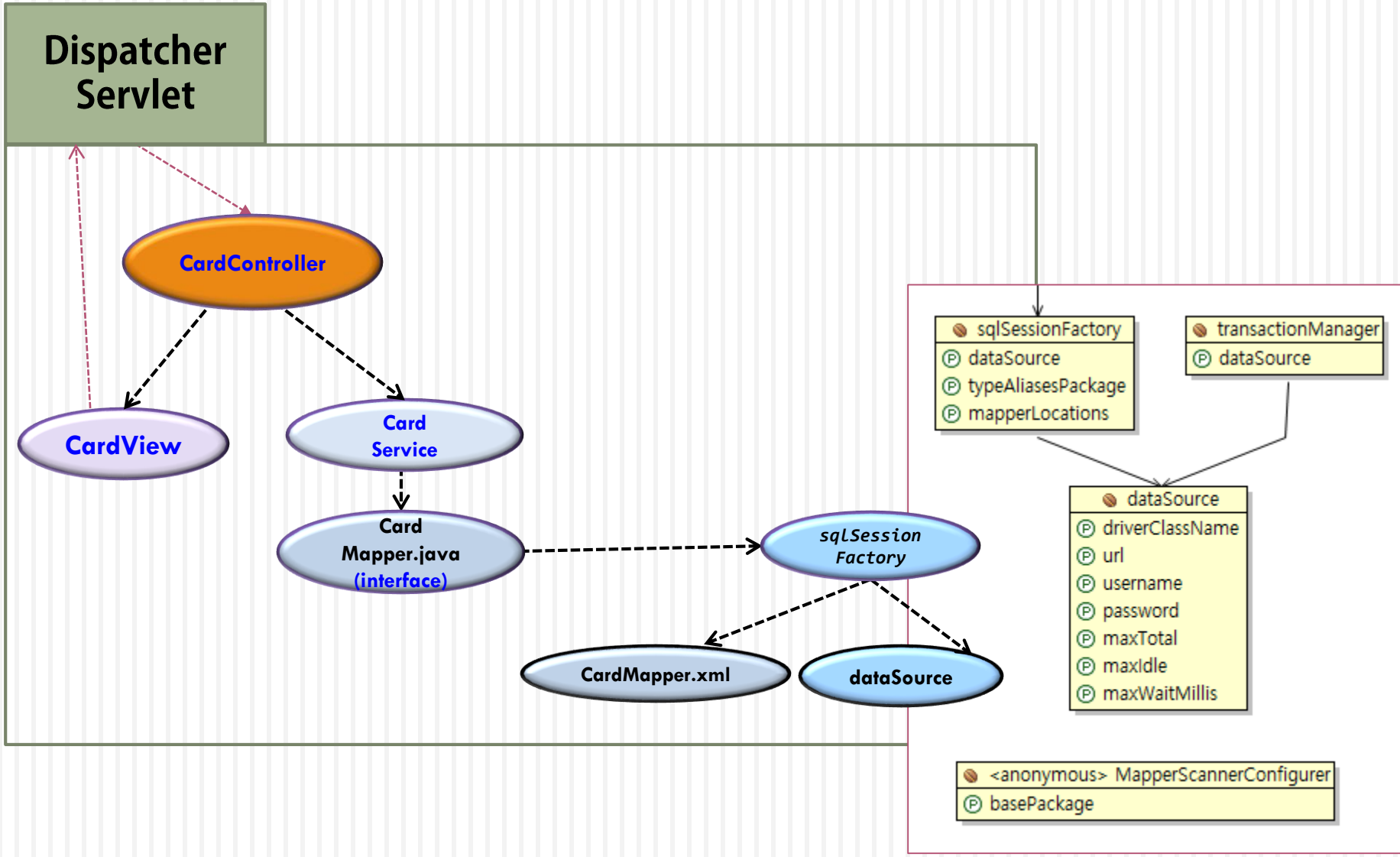
web.xml 구성

```
<!-- utf-8 filter -->
<filter>
    <filter-name>encodingFilter</filter-name>
    <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
    <init-param>
        <param-name>encoding</param-name>
        <param-value>UTF-8</param-value>
    </init-param>
    <init-param>
        <param-name>forceEncoding</param-name>
        <param-value>true</param-value>
    </init-param>
</filter>
<filter-mapping>
        <filter-name>encodingFilter</filter-name>
        <url-pattern>/*</url-pattern>
</filter-mapping>
<!-- The definition of the Root Spring Container shared by all Servlets and Filters -->
<context-param>
-----------------------------------------------------------------------
<servlet-mapping>
   <servlet-name>appServlet</servlet-name>
   <url-pattern>/</url-pattern>
</servlet-mapping>
 <servlet-mapping>
    <servlet-name>default</servlet-name>
    <url-pattern>*.css</url-pattern>
    <url-pattern>*.js</url-pattern>
    <url-pattern>*.gif</url-pattern>
    <url-pattern>*.jpg</url-pattern>
 </servlet-mapping>
```

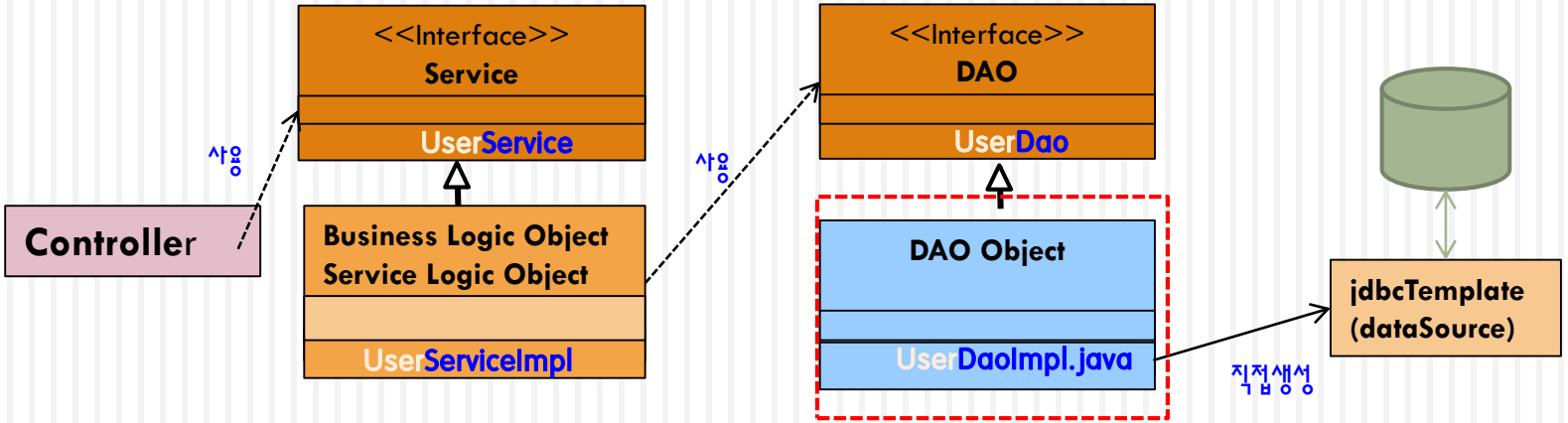
로그인 MVC 처리흐름-

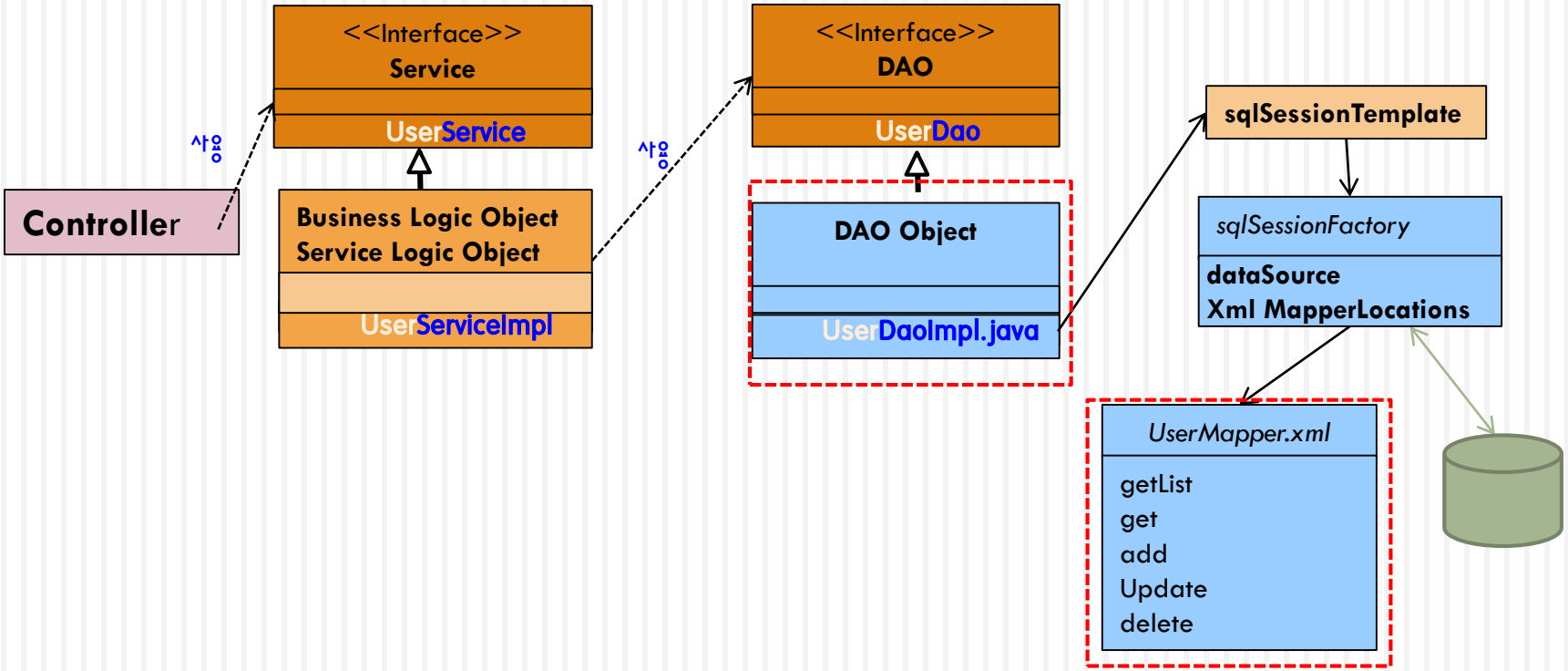

http:// login4mvn/login.html
Dispa
tcher
Servlet
컨트롤러 선택
요청 ②
HandlerMapping
③ 처리위임
ModelAndView
LoginController
④
Model
UserService
UserDao
jdbcTemplate
DataSource
⑤ 뷰 선택요청
ViewResolver
ModelAndView
⑥ 결과물 출력
<form action="login.html">
login.jsp
loginSuccess.jsp

MVC 모델과 IOC 컨테이너 객체 구성

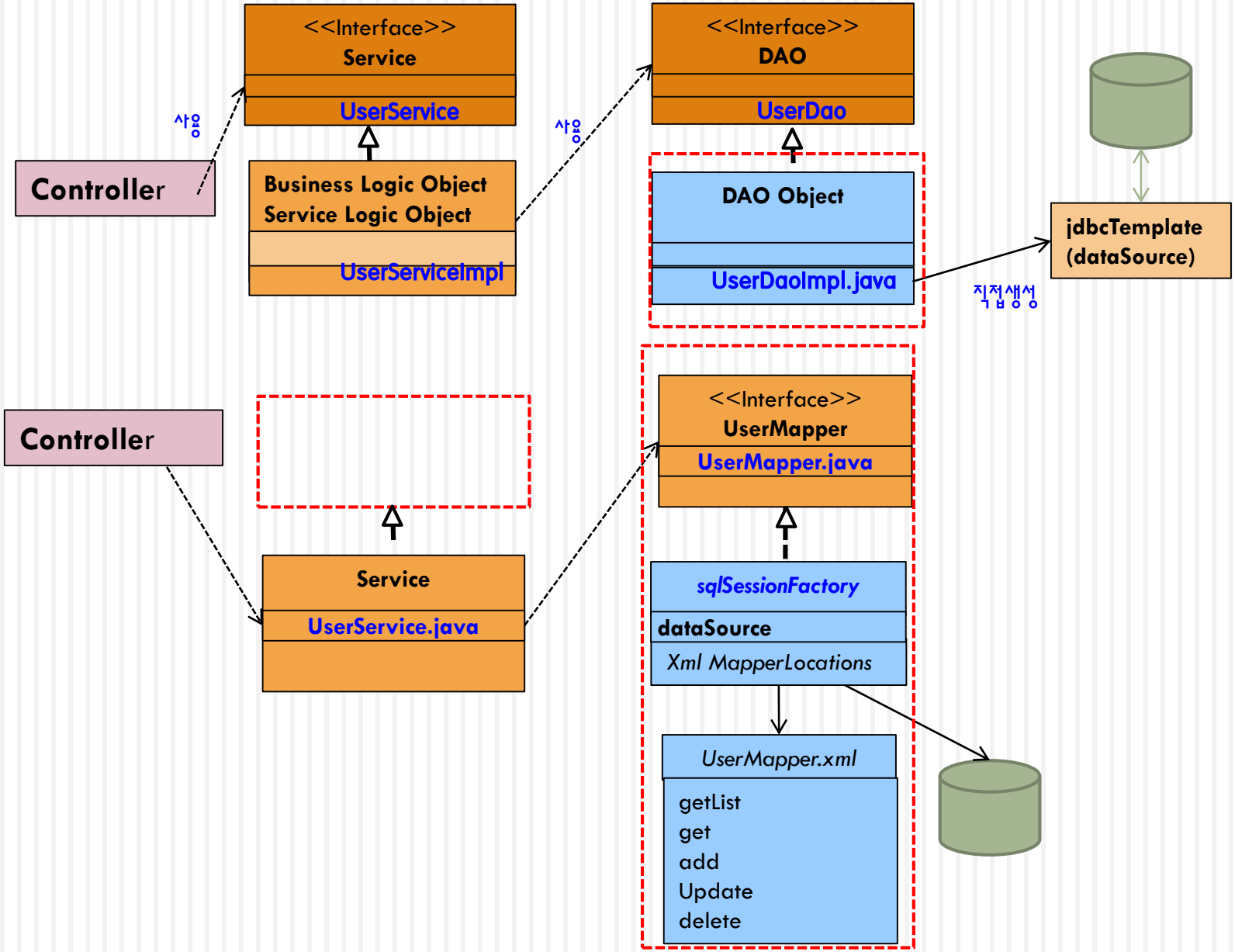


MyBatis Mapper를 이용한 Service 클래스와 DAO 객체 구현

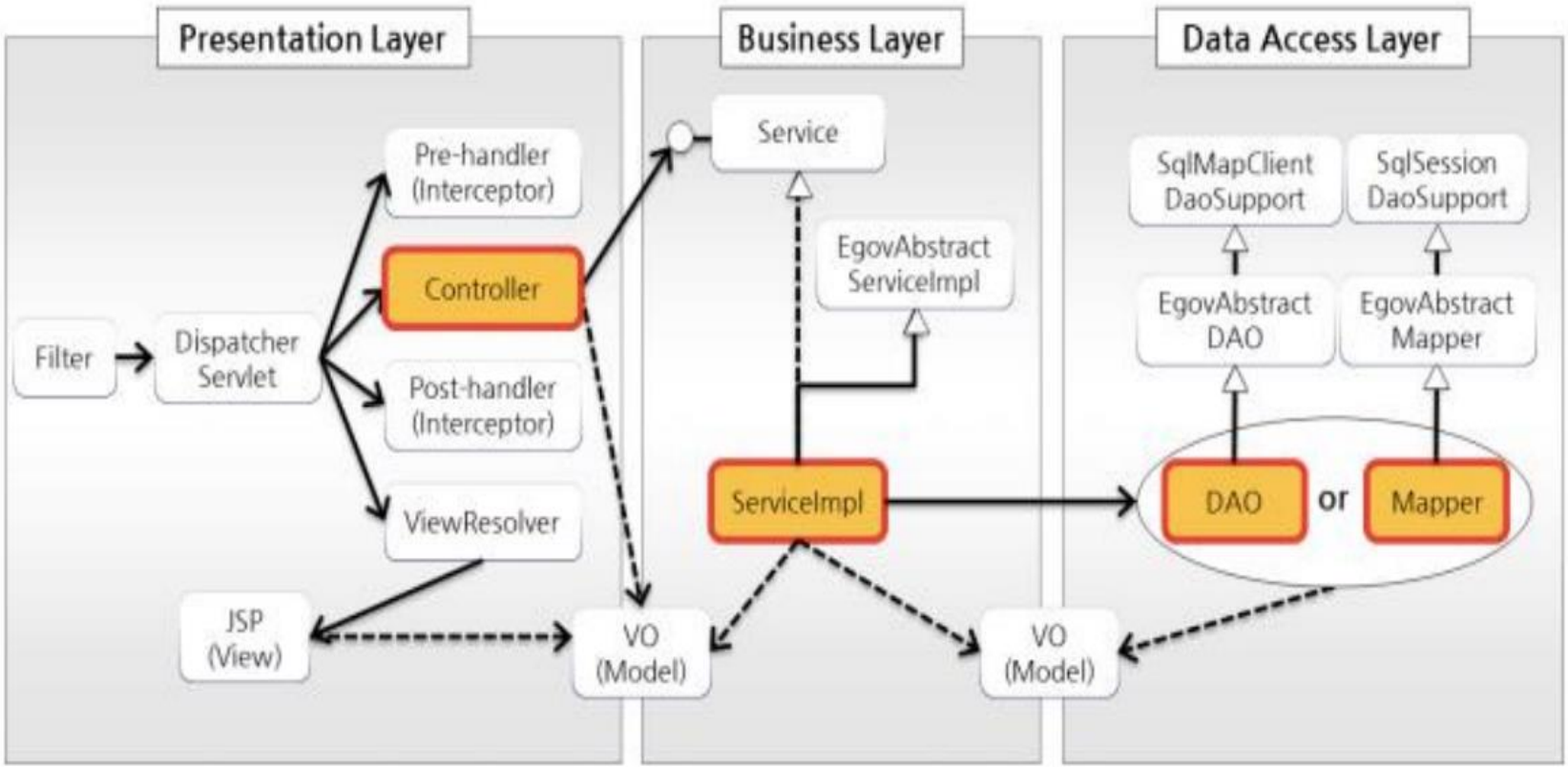




MyBatis Mapper를 이용한 Service 클래스와 DAO 객체 구현



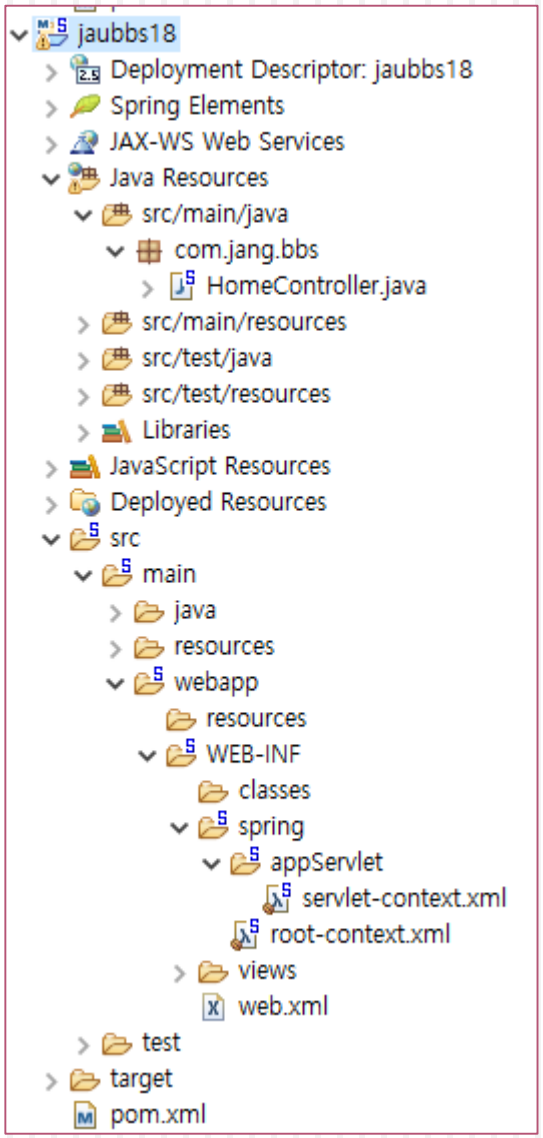
전자정부 표준프레임워크

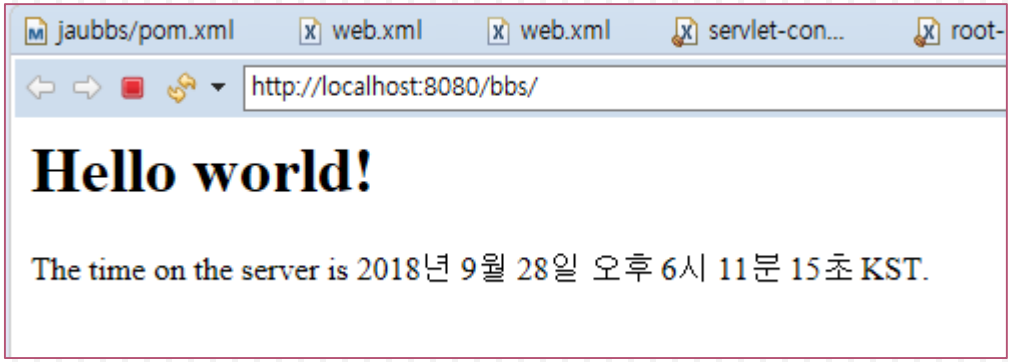


@Controler, @Service, @Repository

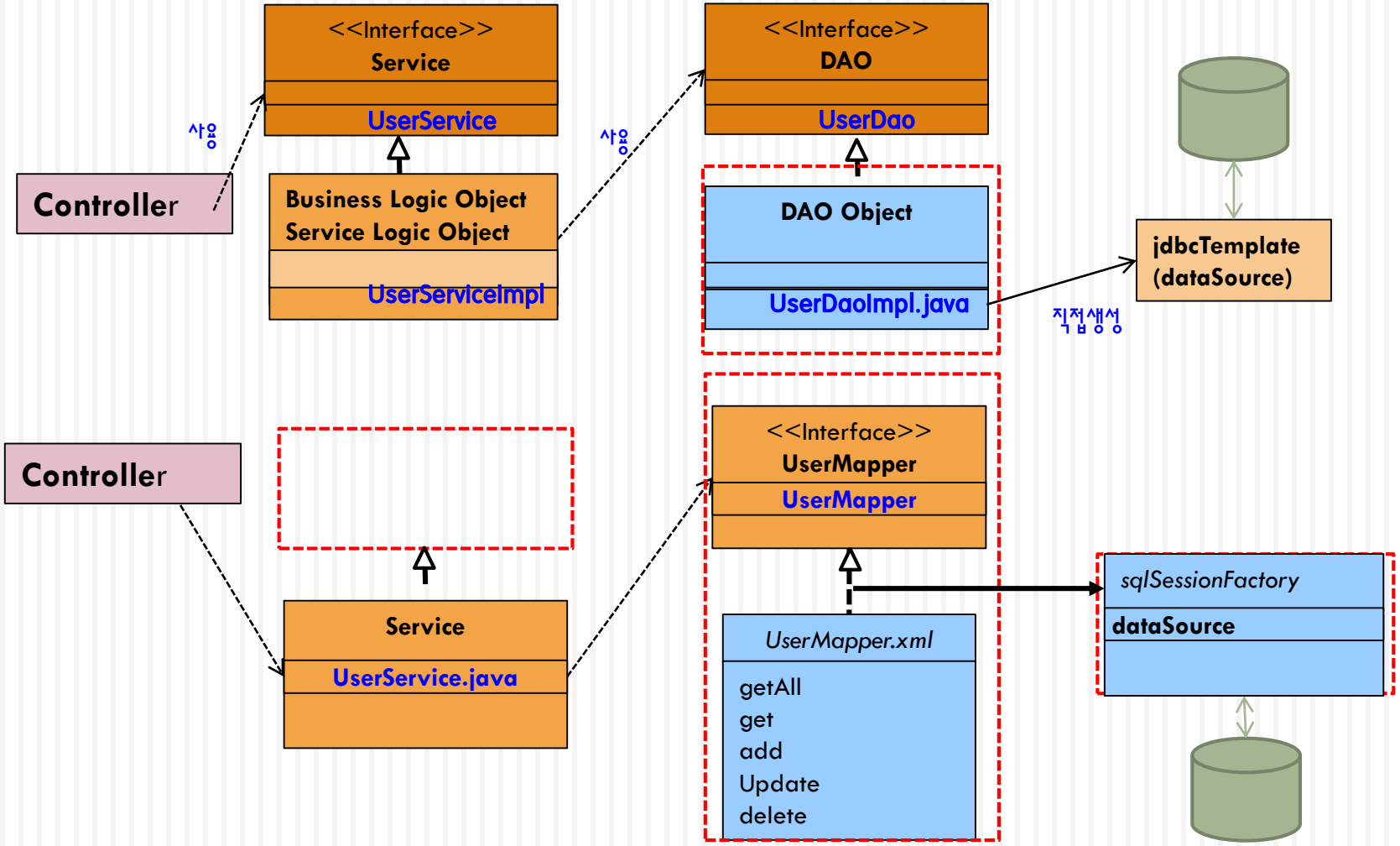
@Mapper

중간 실행 확인


jaubbs18
Deployment Descriptor: jaubbs18
Spring Elements
JAX-WS Web Services
Java Resources
src/main/java
com.jang.bbs
HomeController.java
src/main/resources
src/test/java
src/test/resources
Libraries
JavaScript Resources
Deployed Resources
src
main
java
resources
webapp
resources
WEB-INF
classes
spring
appServlet
servlet-context.xml
root-context.xml
views
web.xml
test
target
pom.xml


jaubbs/pom.xml
web.xml
web.xml
servlet-con...
root-
http://localhost:8080/bbs/
Hello world!
The time on the server is 2018년 9월 28일 오후 6시 11분 15초 KST.

MyBatis Mapper를 이용한 Service 클래스와 DAO 객체 구현



도메인 분석

- 도메인(domain) - 영토, 분야, 영역, 범위를 뜻하는 단어
  - 인터넷 주소의 의미도 확장해서 쓰임
  - 어떤 전문 분야에 대해 얘기할 때 ' 해당 도메인에 대한 지식이 필요' 하다라는 표현

- 도메인 분석
  - 기존 시스템, 개발 이력, 해당분야에 대한 전문 지식을 기반으로 도메인 관련 정보를 수집, 구조화하여 표현하는 과정
  - 대부분의 도메인 분석의 결과는 통일된 방식이 없이 도식적으로 표현되고 수작업으로 작성

- 배경지식
  - 어떤 일을 하거나 연구할 때, 이미 머릿속에 들어 있거나 기본적으로 필요한 지식

- 보고
  - 여러 분야의 비즈니스 활동이 어떻게 진행되고 있는지를 모니터하기 위해 서로 다른 종류의 데이터를 쉽게 알아볼 수 있는 요약된 형태로 정리하는 작업

- 분석
  - 비즈니스 활동을 이해하고 비즈니스 성과를 개선하기 위해 필요한 것이 무엇인지를 찾기위해 데이터와 자료들을 더 깊이 들여다보는 작업

- 참고예제
  - https://docs.Microsoft.com/ko-kr/azure/architecture/microservices/domain-analysis

스마트형 게시판 도메인 분석

https://www.cosmosfarm.com/threads/questions/10

KBoard 커뮤니티

추천	답변	조회		글쓴이	작성일
0	0	12	게시판에 글이 안올라갑니다 안녕하세요. 게시판에 글이 안올라가네요. 최신버전으로 업데이트 했어요. 업데이트 전에 갑자기 k보드 미디어추가시 이미지화일이 첨부가 안되어서 업데이트를 ...	마이꼴팡	2018.10.06 21:51
0	1	36	좀 작게 해서 나오게 가능...	박회장	2018.10.05 14:20
0	2	34	알고 싶습니다. [카...	박회장	2018.10.05 10:30
0	1	40			

게시판에 글이 안올라갑니다

안녕하세요.

게시판에 글이 안올라가네요. 최신버전으로 업데이트 했어요.

업데이트 전에 갑자기 k보드 미디어추가시 이미지화일이 첨부가 안되어서 업데

그러더니 게시판에 글쓰기 기능은 되는데 올라가지 않습니다.

빠른 시일내에 알려주시면 감사하겠습니다.

전화 010-9071-7534

마이꼴팡 (1 Point) · 2018.10.06 21:51 · 조회 24

추천 0 비추천

추천 순서 최신 순서 오래된 순서

스레드봇 (12022 Point) · 2018.10.08 08:58

비회원으로도 테스트해볼 수 있는 페이지 주소도 알려주시겠어요?

고맙습니다.

추천 0 비추천 기부하기 수정하기 삭제

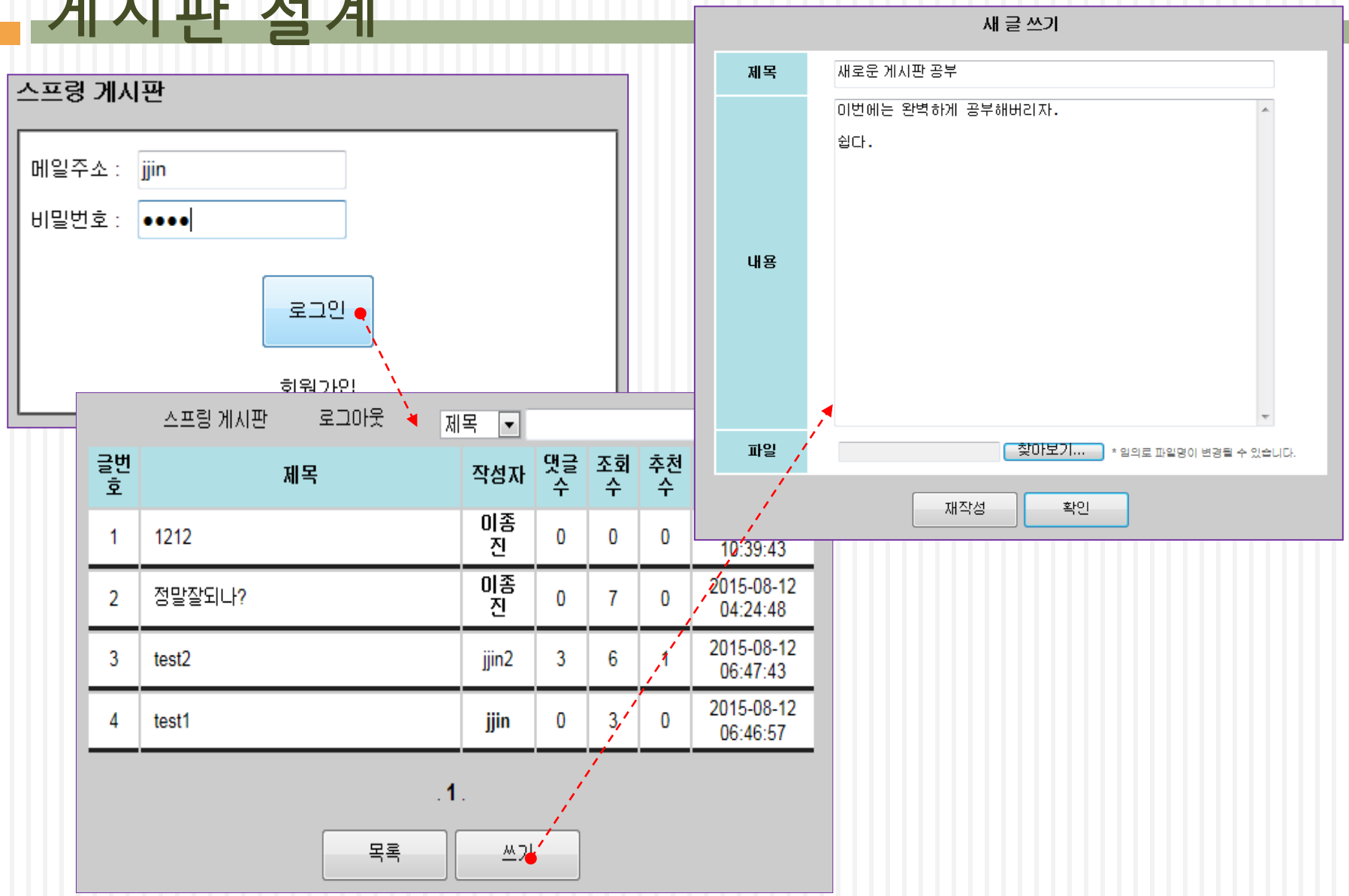
이메일 *

사용자명 *

소스 스타일 문단

답변 달기

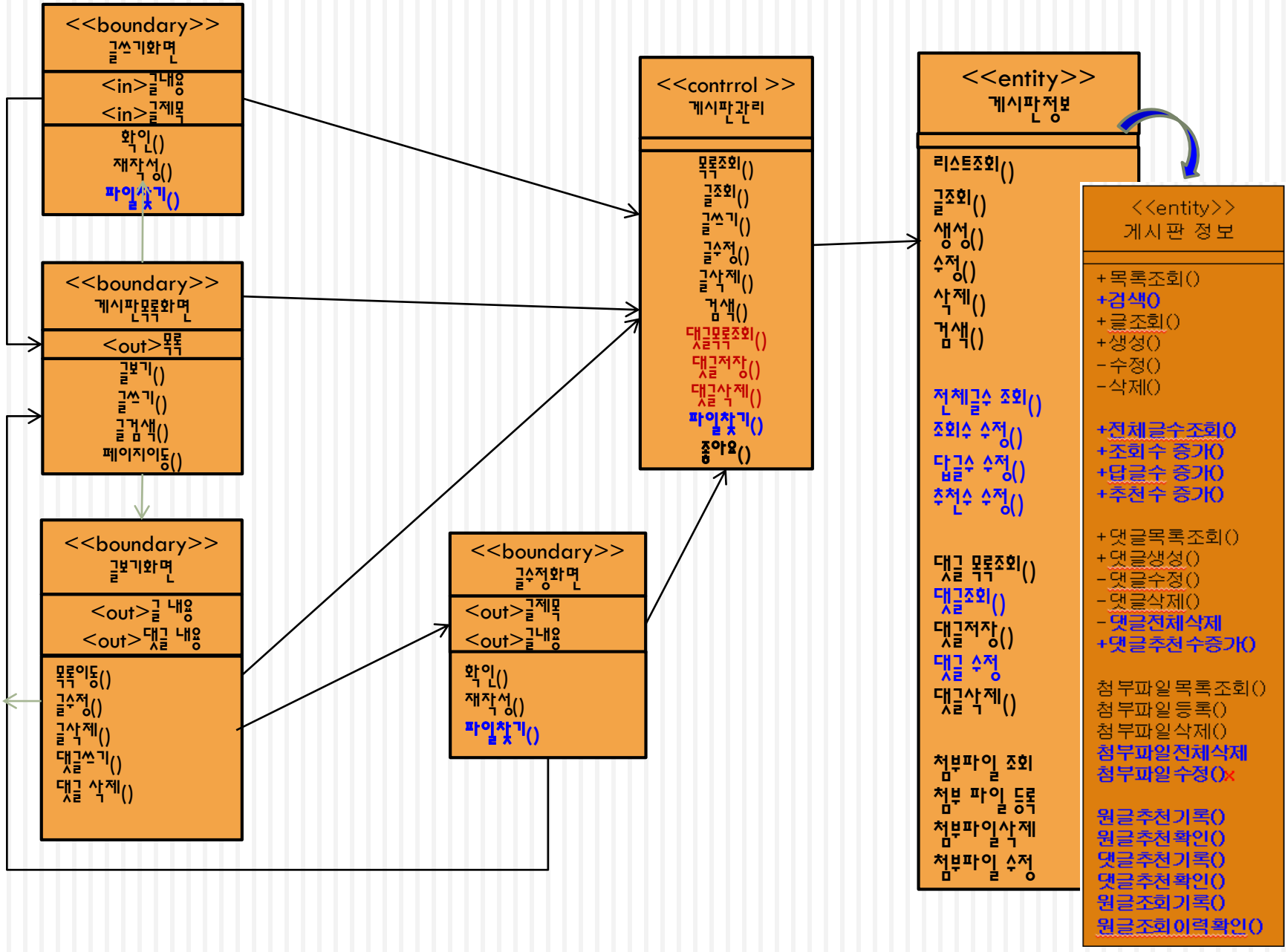
게시판 설계


스프링 게시판
메일주소 :
jjin
비밀번호 :
로그인
회원가입
스프링 게시판
로그아웃
제목
글번호
제목
작성자
댓글수
조회수
추천수
1
1212
이종진
0
0
0
10:39:43
2
정말잘되나?
이종진
0
7
0
2015-08-12 04:24:48
3
test2
jjin2
3
6
1
2015-08-12 06:47:43
4
test1
jjin
0
3
0
2015-08-12 06:46:57
1
목록
쓰기
새 글 쓰기
제목
새로운 게시판 공부
내용
이번에는 완벽하게 공부해버리자.
쉽다.
파일
찾아보기...
* 임의로 파일명이 변경될 수 있습니다.
재작성
확인

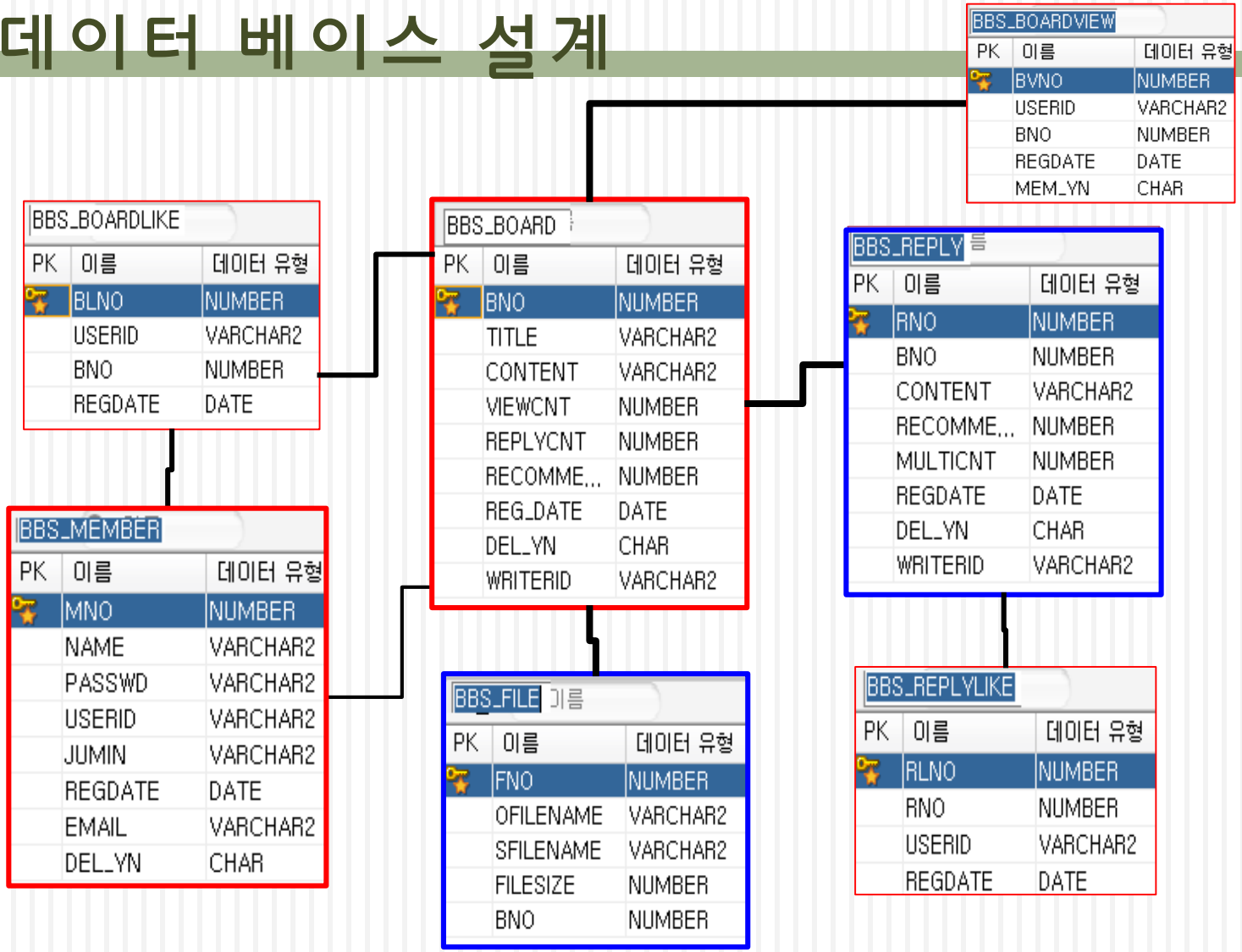
글보기, 삭제

스프링 게시판
로그아웃
제목
검색
글번호
제목
작성자
댓글
조회
추천
작성일
1
1212
2
정말잘되나?
3
test2
4
test1
글보기
정말잘되나?
작성자
조회수
추천수
이종진
8
0
2015-08-1
내용
ewrewrwerweewrwerwe
댓글
이종진
댓글 삭제
2015-08-23 17:57:09.0
가나다라
댓글 쓰기
확인
삭제
수정
목록
글수정
제목
정말잘되나?
내용
수정할수 있습니다. 잘됩니다.
파일
찾아보기...
업로드된 파일:
재작성
확인

게시판 클래스 다이어그램



데이터 베이스 설계



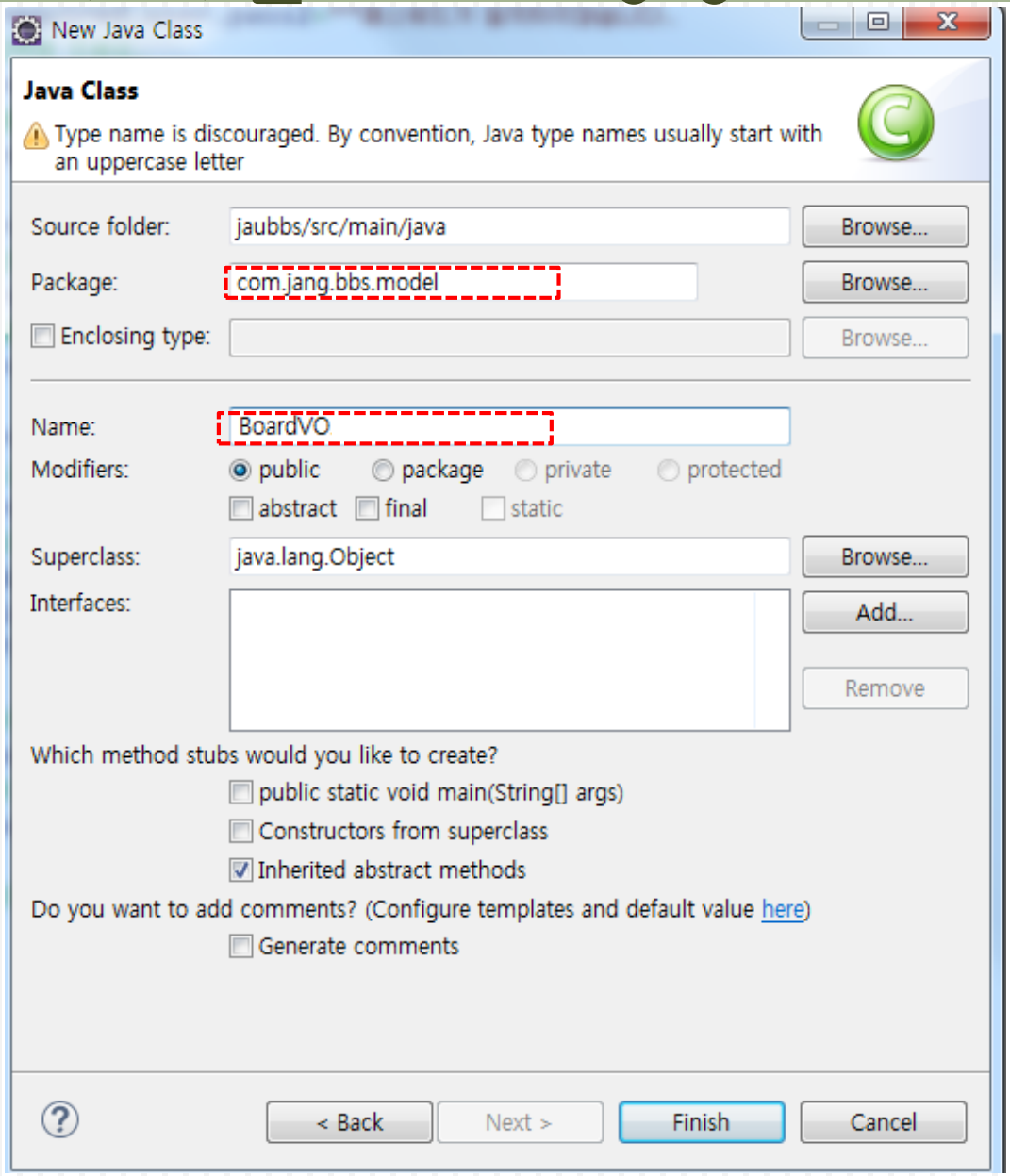
외부키 (foreign key) – 한 테이블의 primary key 를 다른 테이블의 속성으로 사용하는 것
한 테이블의 속성이 다른 테이블의 primary key 값을 갖는 경우.

테이블 구성(2)

```
CREATE TABLE BBS_REPLYLIKE (
        RLNO NUMBER PRIMARY KEY ,
        RNO NUMBER NOT NULL,
        USERID VARCHAR2(20 BYTE) NOT NULL,
        REGDATE DATE DEFAULT sysdate
);

CREATE TABLE BBS_BOARDLIKE (
        BLNO NUMBER PRIMARY KEY ,
        USERID VARCHAR2(20 BYTE) NOT NULL,
        BNO NUMBER NOT NULL,
        REGDATE DATE DEFAULT sysdate
)

CREATE TABLE BBS_BOARDVIEW (
        BVNO NUMBER PRIMARY KEY ,
        USERID VARCHAR2(20 BYTE) NOT NULL,
        BNO NUMBER NOT NULL,
        REGDATE DATE DEFAULT sysdate,
        MEM_yn char(1) default 'y' check (mem_yn in ('y','n'))

)
```

```
create table bbs_board (
        bno number primary key,
        title varchar2(500) not null,
        content varchar2(4000) not null,
        viewcnt number default 0,
        replycnt number default 0,
        recommendcnt number default 0,
        writerID varchar2(50) not null,
        regdate date default sysdate,
        del_yn char(1) default 'n' check (del_yn in ('y','n'))
);
create table bbs_reply (
        rno number primary key,
        bno number not null,
        content varchar2(2000) not null,
        recommendcnt number default 0,
        multicnt number default 0,
        writerID varchar2(50) not null,
        regdate date default sysdate,
        del_yn char(1) default 'n' check (del_yn in ('y','n'))
);
create table bbs_file (
        fno number primary key,
        bno number not null,
        ofilename varchar2(1000 byte) not null,
        sfilename varchar2(1000 byte) not null,
        filesize number not null
);
create table bbs_member (
        mno number primary key,
        userid varchar2(30) not null unique,
        passwd varchar2(30) not null,
        name   varchar2(100) not null,
        jumin   varchar2(14)  not null unique,
        email varchar2(100) not null unique,
        regdate date default sysdate,
        del_yn char(1) default 'n' check (del_yn in ('y','n'))
);
```

DTO(VO) 클래스의 생성



Lombok - @Data, Validator (입력하지 말것)

```java
package com.jang.bbs.model;

import javax.validation.constraints.NotEmpty;

public class UserVO {


    private int mno;

    @NotEmpty(message="id를 입력하세요!")
    private String userId;
    @NotEmpty(message="패스워드를 입력하세요!")
    private String passwd;
    private String name;
    private String jumin;
    private String regDate;
    private String email;
    private char del_yn='n';

    public int getMno() {
        return mno;
    }
    public void setMno(int mno) {
        this.mno = mno;
    }
```

Gatter & Setter 삽입

```java
package com.jang.bbs.model;

import lombok.Data;

@Data
public class BoardVO {

    private int bno;
    private String title;
    private String content;
    private int viewcnt = 0;
    private int replycnt=0;
    private int recommendcnt=0;
    private String writerId;
    private String regDate;
    private char del_yn='n';

}
```

Lombok 사용

AttFile.java
Reply.java 에도 추가

```pom.xml
<dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <version>1.18.2</version>
    <scope>provided</scope>
</dependency>
```

DTO 클래스 작성(2)

BBS_MEMBER

PK	이름	데이터 유형
	MNO	NUMBER
	NAME	VARCHAR2
	PASSWD	VARCHAR2
	ID	VARCHAR2
	JUMIN	VARCHAR2
	REG_DATE	DATE
	EMAIL	VARCHAR2
	DEL_YN	CHAR

UserVO.java

```
public class UserVO {
  private int mno;
  private String name;
  private String passwd;
  private String userId;
  private String jumin;
  private String email;
  private String regDate;
  private char del_yn='n';
}
```

Gatter & Setter 삽입

BBS_REPLY

PK	이름	데이터 유형
	RNO	NUMBER
	BNO	NUMBER
	CONTENT	VARCHAR2
	RECOMME...	NUMBER
	MULTICNT	NUMBER
	REGDATE	DATE
	DEL_YN	CHAR
	WRITERID	VARCHAR2

ReplyVO.java

```
private int rno;
private int bno;
private String content;
private int recommendcnt=0;
private int multicnt=0;
private String writerId;
private String regDate;
private char del_yn='n';
```

BBS_BOARD

PK	이름	데이터 유형
	BNO	NUMBER
	TITLE	VARCHAR2
	CONTENT	VARCHAR2
	VIEWCNT	NUMBER
	REPLYCNT	NUMBER
	RECOMME...	NUMBER
	REG_DATE	DATE
	DEL_YN	CHAR
	WRITERID	VARCHAR2

BoardVO.java

```
private int bno;
private String title;
private String content;
private int viewcnt = 0;
private int replycnt=0;
private int recommendcnt=0;
private String writerId;
private String regDate;
private char del_yn='n';
```

BBS_FILE

PK	이름	데이터 유형
	FNO	NUMBER
	OFILENAME	VARCHAR2
	SFILENAME	VARCHAR2
	FILESIZE	NUMBER
	BNO	NUMBER

AttFileVO.java

```
private int fno;
private int bno;
private String ofilename;
private String sfilename;
private long filesize;
```

DTO 클래스 작성(2)

```
public class BoardLikeVO {

      private int blno;
      private int bno;
      private String userId;
      private String regDate;
}
```

BBS_BOARDLIKE

PK	이름	데이터 유형
	BLNO	NUMBER
	USERID	VARCHAR2
	BNO	NUMBER
	REGDATE	DATE

```
public class ReplyLikeVO {

      private int rlno;
      private int rno;
      private String userId;
      private String regDate;

}
```

BBS_REPLYLIKE

PK	이름	데이터 유형
	RLNO	NUMBER
	RNO	NUMBER
	USERID	VARCHAR2
	REGDATE	DATE

```
public class BoardViewVO {
      private int bvno;
      private int bno;
      private String userId;
      private char mem_yn;
      private String regDate;
}
```

BBS_BOARDVIEW

PK	이름	데이터 유형
	BVNO	NUMBER
	USERID	VARCHAR2
	BNO	NUMBER
	REGDATE	DATE
	MEM_YN	CHAR

Get 방식의 url 전송과 XML 파일에 전달하는 매개변수를 줄이기 위한 전달객체 생성

```
public class SearchVO {
      private String type = "";
      private String keyword = "";
      private int    page=1;
      private int    startRow=1;
      private int    endRow=1;
}
```

Mybatis 에서 복수 파라메터 사용

- @Param 이용

```
List<Board> getBoardList(@Param(value = "startRow")int startRow, @Param(value = "endRow")int endRow);
```

Mapper Interface

```
<select id="getBoardList"      resultType="Board" >  </select>
```

Mapper xml

```
<select id="getBoardList" parameters="int startRow; int endRow" resultType="Board" >  </select>
```

생략가능

- Hash Map 사용

```
HashMap<String, Object> hmap = new HashMap<String, Object>();
hmap.put("startRow", startRow);
hmap.put("endRow", endRow);

List<Board> list = boardMapper.getBoardList(hmap);
```

Service

```
List<Board> getBoardList(HashMap<String, Object> hmap);
```

Mapper Interface

```
<select id="getBoardList" parameterType="HashMap" resultType="Board" >
```

Mapper xml

생략가능

Mybatis 에서 복수 파라메터 사용

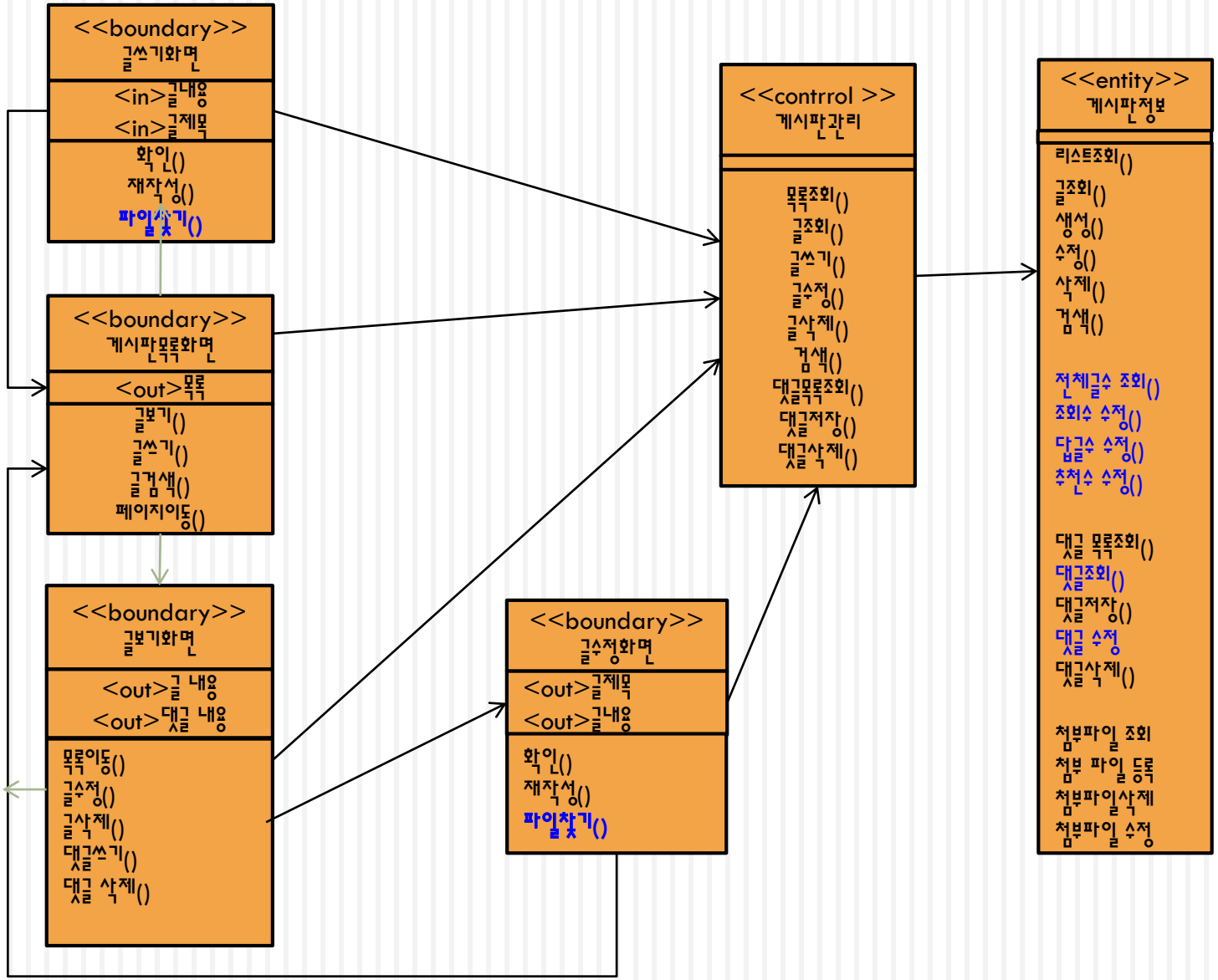
- DTO (전달객체)사용

```
public class SearchVO {
      private String type = "";
      private String keyword = "";
      private int   page=1;
      private int   startRow=1;
      private int   endRow=1;
}
```

```
List<Board> list = boardMapper.getBoardList(searchVo);
```

```
<select id="getBoardList" parameterType="SearchVO" resultType="Board" >
     <![CDATA[
     SELECT  *
     FROM  ( SELECT ROWNUM AS rnum, a.*
             FROM ( SELECT bno, title, writer, replycnt, viewcnt, regdate, recommendcnt
                    FROM tbl_board order by bno desc ) a
             WHERE ROWNUM <= #{endRow})
     WHERE rnum >= #{startRow}
     ]]>
</select>
```

게시판 클래스 다이어그램



BoardMapper interface 설계

```java
@Mapper
public interface BoardMapper {

    List<BoardVO> getBoardList(SearchVO searchVO); //글 목록조회
    BoardVO getArticle(int bno);          //글 조회
    int writeArticle(BoardVO board);      //글저장
    int updateArticle(BoardVO board);     //글수정
    void deleteArticle(int bno);          //글삭제

    int getTotalRow(SearchVO searchVO); //전체글 수 조회

    int incrementViewCnt(int bno);   //조회수 증가
    int incrementGoodCnt(int bno);   //좋아요 증가
    int incrementReplyCnt(int bno); //댓글 수 증가

    List<ReplyVO> getReplyList(int bno);   //댓글 목록조회
    ReplyVO getReply(int rno);   // 댓글조회

    int writeReply(ReplyVO reply); //댓글저장
    int updateReply(ReplyVO reply); //댓글수정
    void deleteReply(int rno);        //댓글삭제
    void deleteReplyBybno(int bno); //원글에 소속된 댓글 전체 삭제

    int incReplyGoodCnt(int rno); //댓글 좋아요 증가
    // int incrementReMultiCnt(int bno);

    List<AttFileVO> getFileList( int bno); //첨부파일 목록 조회
    String getFileName(int fno);   //첨부파일 이름 조회
    int insertFile(AttFileVO file); //첨부파일 저장

    void deleteFile(int fno);   //첨부파일 삭제
    void deleteFileBybno(int bno); //원글에 소속된 첨부파일 전체 삭제

    int addBoardLike(BoardLikeVO boardLike); //id-특정글 좋아요 기록 저장
    int getBoardLike(BoardLikeVO boardLike); //id-특정글 좋아요 기록 조회

    int addReplyLike(ReplyLikeVO replyLike); //id-특정댓글 좋아요 기록 저장
    int getReplyLike(ReplyLikeVO replyLike); //id-특정댓글 좋아요 기록 조회

    int addBoardView(BoardViewVO boardView); //id-특정글  조회이력  기록
    int getBoardView(BoardViewVO boardView); //id-특정글  조회이력  읽기
```

<<entity>>
게시판 정보

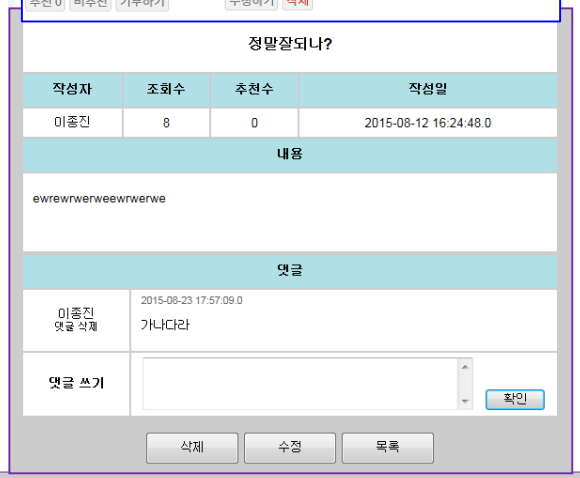
+목록조회()
+검색()
+글조회()
+생성()
-수정()
-삭제()

+전체글수조회()
+조회수 증가()
+답글수 증가()
+추천수 증가()

+댓글목록조회()
+댓글생성()
-댓글수정()
-댓글삭제()
-댓글전체삭제
+댓글추천수증가()

첨부파일목록조회()
첨부파일등록()
첨부파일삭제()
첨부파일전체삭제
첨부파일수정()x

원글추천기록()
원글추천확인()
댓글추천기록()
댓글추천확인()
원글조회기록()
원글조회이력확인()



BoardMapper interface 작성

```java
@Mapper
public interface BoardMapper {

    List<BoardVO> getBoardList(SearchVO searchVO); //글 목록조회
    BoardVO getArticle(int bno);          //글 조회
    int writeArticle(BoardVO board);      //글저장
    int updateArticle(BoardVO board);     //글수정
    void deleteArticle(int bno);          //글삭제

    int getTotalRow(SearchVO searchVO); //전체글 수 조회

    int incrementViewCnt(int bno);   //조회수 증가
    int incrementGoodCnt(int bno);   //좋아요 증가
    int incrementReplyCnt(int bno); //댓글 수 증가

    List<ReplyVO> getReplyList(int bno);   //댓글 목록조회
    ReplyVO getReply(int rno);   // 댓글조회

    int writeReply(ReplyVO reply); //댓글저장
    int updateReply(ReplyVO reply); //댓글수정
    void deleteReply(int rno);        //댓글삭제
    void deleteReplyBybno(int bno); //원글에 소속된 댓글 전체 삭제

    int incReplyGoodCnt(int rno); //댓글 좋아요 증가
    // int incrementReMultiCnt(int bno);

    List<AttFileVO> getFileList( int bno); //첨부파일 목록 조회
    String getFileName(int fno);   //첨부파일 이름 조회
    int insertFile(AttFileVO file); //첨부파일 저장

    void deleteFile(int fno);   //첨부파일 삭제
    void deleteFileBybno(int bno); //원글에 소속된 첨부파일 전체 삭제

    int addBoardLike(BoardLikeVO boardLike); //id-특정글 좋아요 기록 저장
    int getBoardLike(BoardLikeVO boardLike); //id-특정글 좋아요 기록 조회

    int addReplyLike(ReplyLikeVO replyLike); //id-특정댓글 좋아요 기록 저장
    int getReplyLike(ReplyLikeVO replyLike); //id-특정댓글 좋아요 기록 조회

    int addBoardView(BoardViewVO boardView); //id-특정글 조회이력 기록
    int getBoardView(BoardViewVO boardView); //id-특정글 조회이력 읽기
```

<<entity>>
게시판 정보

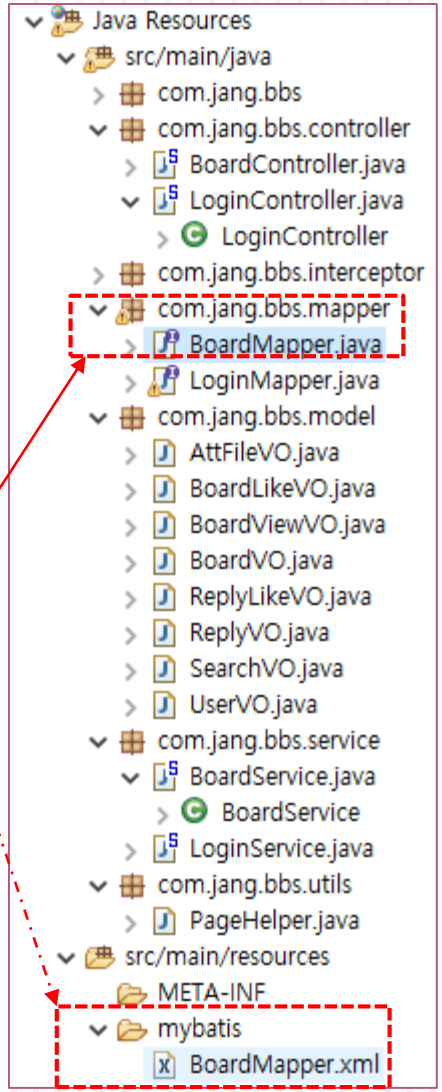
+목록조회()
+검색()
+글조회()
+생성()
-수정()
-삭제()

+전체글수조회()
+조회수 증가()
+답글수 증가()
+추천수 증가()

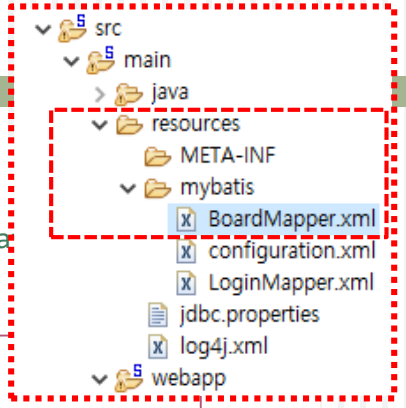
+댓글목록조회()
+댓글생성()
-댓글수정()
-댓글삭제()
-댓글전체삭제
+댓글추천수증가()

첨부파일목록조회()
첨부파일등록()
첨부파일삭제()
첨부파일전체삭제

원글추천기록()
원글추천확인()
댓글추천기록()
댓글추천확인()
원글조회기록()
원글조회이력확인()



BoardMapper.xml 작성




```xml
<?xml version="1.0" encoding="UTF-8"?>

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

<mapper namespace="com.jang.bbs.mapper.BoardMapper">
 <select id="getBoardList"  parameterType="SearchVO" resultType="BoardVO" >

   SELECT  bno, title, writerId, replycnt, viewcnt, to_char(regDate, 'YYYY-MM-DD HH24:MI') e
          regDate, recommendcnt
   FROM  (  SELECT ROWNUM AS rnum, a.*
           FROM   ( SELECT * FROM   bbs_board  WHERE  del_yn='n' order by bno desc  ) a
          )
   WHERE rnum  between #{startRow} and #{endRow}

 </select>
```

```xml
<select id="getTotalRow" parameterType="SearchVO" resultType="int">
   SELECT count(*) AS totalrow
   FROM bbs_board
   WHERE del_yn='n'
</select>
```

```xml
</mapper>
```

```java
List<BoardVO> getBoardList(SearchVO searchVO);

BoardVO getArticle(int bno);       //글조회
int writeArticle(BoardVO board); //글저장
int updateArticle(BoardVO board); //글수정
void deleteArticle(int bno);      //글삭제

int getTotalRow(SearchVO searchVO); //전체 글수 조
```

```java
public class SearchVO {
    private String type = "";
    private String keyword = "";
    private int   page=1;
    private int   startRow=1;
    private int   endRow=1;
}
```

BoardMapper.xml 작성

```
<select id="getArticle" resultType="BoardVO">
    SELECT  bno , title, content, writerId, to_char(regDate, 'YYYY-MM-DD HH24:MI') regDate, viewcnt,
            replycnt, recommendcnt
    FROM  bbs_board
    WHERE  bno = #{bno}
</select>

<insert id="writeArticle" keyProperty="bno">
   <selectKey keyProperty="bno" resultType="int" order="BEFORE">
       SELECT COALESCE(MAX(bno), 0) + 1 AS bno  FROM  bbs_board
   </selectKey>

   INSERT  INTO  bbs_board (bno, title, content, writerId) VALUES ( #{bno}, #{title}, #{content}, #{writerId} )
</insert>
```

```
List<BoardVO> getBoardList(SearchVO searchVO); //글 목록 조회

BoardVO getArticle(int bno);       //글조회
int writeArticle(BoardVO board); //글저장
int updateArticle(BoardVO board); //글수정
void deleteArticle(int bno);      //글삭제

int getTotalRow(SearchVO searchVO); //전체 글수 조회
```

BoardVO

```
private int bno;
private String title;
private String content;
private int viewcnt = 0;
private int replycnt=0;
private int recommendcnt=0;
private String writerId;
private String regDate;
private char del_yn='n';
```

BS_BOARD

PK	이름	데이터 유형
	BNO	NUMBER
	TITLE	VARCHAR2
	CONTENT	VARCHAR2
	VIEWCNT	NUMBER
	REPLYCNT	NUMBER
	RECOMME...	NUMBER
	REG_DATE	DATE
	DEL_YN	CHAR
	WRITERID	VARCHAR2

BoardMapper.xml 작성

```
<update id="updateArticle">
    UPDATE  bbs_board
    SET  title =  #{title}, content = #{content}, writerId = #{writerId}, viewcnt = #{viewcnt}
    WHERE  bno  = #{bno}
</update>

<delete id="deleteArticle">
    DELETE  FROM  bbs_board
    WHERE  bno = #{bno}
</delete>

<update id="incrementViewCnt">
    UPDATE  bbs_board
    SET     viewcnt = viewcnt+1
    WHERE  bno  = #{bno}
</update>

<update id="incrementGoodCnt">
    UPDATE  bbs_board
    SET     recommendcnt=recommendcnt+1
    WHERE   bno  = #{bno}
</update>

<update id="incrementReplyCnt">
    UPDATE  bbs_board
    SET     replycnt = replycnt + 1
    WHERE   bno  = #{bno}
</update>
```

```
List<BoardVO> getBoardList(SearchVO searchVO);

BoardVO getArticle(int bno);       //글조회
int writeArticle(BoardVO board);  //글저장
int updateArticle(BoardVO board); //글수정
void deleteArticle(int bno);      //글삭제

int getTotalRow(SearchVO searchVO); //전체 글

int incrementViewCnt(int bno);  //조회수 증가
int incrementGoodCnt(int bno);  //좋아요 증가
int incrementReplyCnt(int bno); //댓글수 증가

List<ReplyVO> getReplyList(int bno);  //댓글 목록
ReplyVO getReply(int rno);  // 댓글 조회

int writeReply(ReplyVO reply); //댓글 저장
int updateReply(ReplyVO reply); //댓글 수정
void deleteReply(int rno);      //댓글 삭제
void deleteReplyBybno(int bno); //원글에 소속된 댓글

int incReplyGoodCnt(int rno); //댓글 좋아요 증가
// int incrementReMultiCnt(int bno);

List<AttFileVO> getFileList( int bno); //첨부파일
String getFileName(int fno);  //파일 이름 조회
int insertFile(AttFileVO file); //파일 저장

void deleteFile(int fno);  //첨부 파일 삭제
void deleteFileBybno(int bno); //원본 소속된 첨부파일

int addBoardLike(BoardLikeVO boardLike); //원본
int getBoardLike(BoardLikeVO boardLike); //원본

int addReplyLike(ReplyLikeVO replyLike); //댓글
int getReplyLike(ReplyLikeVO replyLike); //댓글
```

BoardMapper.xml 작성

```xml
<select id="getReplyList" resultType="ReplyVO" >
    SELECT  rno, bno, content, recommendcnt, multicnt, to_char(regDate, 'YYYY-MM-DD HH24:MI') regDate, writerId
    FROM   bbs_reply
    WHERE  bno = #{bno} and del_yn='n'
</select>

<select id="getReply" resultType="ReplyVO">
    SELECT  rno, bno, content, writerId, regDate  FROM   bbs_reply
    WHERE  rno = #{rno}
</select>

<insert id="writeReply" keyProperty="rno">
    <selectKey keyProperty="rno" resultType="int" order="BEFORE">
        SELECT COALESCE(MAX(rno), 0) + 1 AS rno
        FROM   bbs_reply
    </selectKey>

    INSERT INTO bbs_reply ( rno, bno, content, writerId )
    VALUES ( #{rno}, #{bno}, #{content}, #{writerId} )
</insert>

<update id="updateReply">
    UPDATE  bbs_reply
     SET    content = #{content}, writerId = #{writerId}, regDate = #{regDate}
    WHERE  rno  = #{rno}
</update>

<delete id="deleteReply">
    DELETE  FROM  bbs_reply
    WHERE rno = #{rno}
</delete>

<delete id="deleteReplyBybno">
    DELETE  FROM  bbs_reply
    WHERE bno = #{bno}
</delete>

<update id="incReplyGoodCnt">
    UPDATE  bbs_reply
    SET     recommendcnt = recommendcnt + 1
    WHERE   rno  = #{rno}
</update>
```

```java
List<BoardVO> getBoardList(SearchVO searchVO);

BoardVO getArticle(int bno);       //글조회
int writeArticle(BoardVO board); //글저장
int updateArticle(BoardVO board); //글수정
void deleteArticle(int bno);      //글삭제

int getTotalArticle(SearchVO searchVO); //전체 글

int incrementViewCnt(int bno);  //조회수 증가
int incrementGoodCnt(int bno);  //좋아요 증가
int incrementReplyCnt(int bno); //댓글수 증가

List<ReplyVO> getReplyList(int bno);  //댓글 목록
ReplyVO getReply(int rno);  // 댓글 조회

int writeReply(ReplyVO reply); //댓글 저장
int updateReply(ReplyVO reply); //댓글 수정
void deleteReply(int rno);      //댓글 삭제
void deleteReplyBybno(int bno); //원글에 소속된 댓글

int incReplyGoodCnt(int rno); //댓글 좋아요 증가
// int incrementReMultiCnt(int bno);

List<AttFileVO> getFileList( int bno); //첨부파일
String getFileName(int fno);  //파일 이름 조회
int insertFile(AttFileVO file); //파일 저장

void deleteFile(int fno);  //첨부 파일 삭제
void deleteFileBybno(int bno); //원본 소속된 첨부파일

int addBoardLike(BoardLikeVO boardLike); //원본
int getBoardLike(BoardLikeVO boardLike); //원본

int addReplyLike(ReplyLikeVO replyLike); //댓글
int getReplyLike(ReplyLikeVO replyLike); //댓글
```

BoardMapper.xml 작성

```
<select id="getFileList" resultType="AttFileVO" >
    SELECT  fno, bno, ofilename, sfilename, filesize
    FROM   bbs_file
    WHERE  bno = #{bno}
</select>

<select id="getFileName" resultType="String" parameterType="int" >
    SELECT  ofilename
    FROM   bbs_file
    WHERE  fno = #{fno}
</select>

<insert id="insertFile" keyProperty="fno">
    <selectKey keyProperty="fno" resultType="int" order="BEFORE">
        SELECT COALESCE(MAX(fno), 0) + 1 AS fno
        FROM   bbs_file
    </selectKey>

    INSERT INTO bbs_file (fno, bno, ofilename, sfilename, filesize )
    VALUES ( #{fno}, #{bno}, #{ofilename}, #{sfilename}, #{filesize})
</insert>

<delete id="deleteFile">
    DELETE  FROM  bbs_file
    WHERE fno = #{fno}
</delete>

<delete id="deleteFileBybno">
    DELETE  FROM  bbs_file
    WHERE bno = #{bno}
</delete>

<insert id="addBoardLike" keyProperty="blno">
    <selectKey keyProperty="blno" resultType="int" order="BEFORE">
        SELECT COALESCE(MAX(blno), 0) + 1 AS blno
        FROM   bbs_boardlike
    </selectKey>

    INSERT INTO bbs_boardlike (blno, bno, userId )
    VALUES ( #{blno}, #{bno}, #{userId})
</insert>
```

```
List<AttFileVO> getFileList( int bno); //
String getFileName(int fno);  //파일 이름 조
int insertFile(AttFileVO file); //파일 저장

void deleteFile(int fno);  //첨부 파일 삭제
void deleteFileBybno(int bno); //원본 소속된

int addBoardLike(BoardLikeVO boardLike);
```

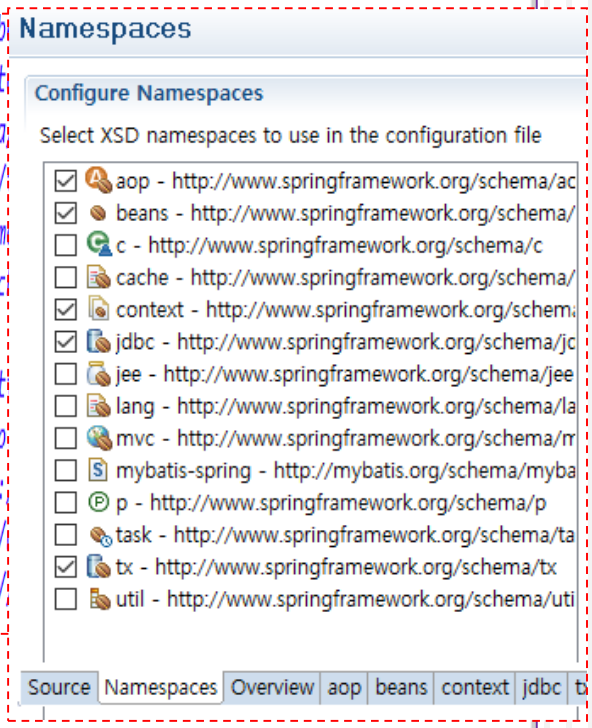
```
int getBoardLike(BoardLikeVO boardLike);

int addReplyLike(ReplyLikeVO replyLike);
int getReplyLike(ReplyLikeVO replyLike);

int addBoardView(BoardViewVO boardView);
int getBoardView(BoardViewVO boardView);
```

```
    <select id="getBoardLike" resultType="int">
        SELECT COALESCE(MAX(bno), 0) AS bno
        FROM   bbs_boardlike
        WHERE  userId = #{userId} AND bno = #{bno}
    </select>

    <insert id="addReplyLike" keyProperty="rlno">
        <selectKey keyProperty="rlno" resultType="int" order="BEFORE">
            SELECT COALESCE(MAX(rlno), 0) + 1 AS rlno
            FROM   bbs_replylike
        </selectKey>

        INSERT INTO bbs_replylike (rlno, rno, userId )
        VALUES ( #{rlno}, #{rno}, #{userId})
    </insert>

    <select id="getReplyLike" resultType="int">
        SELECT COALESCE(MAX(rno), 0) AS rno
        FROM   bbs_replylike
        WHERE  userId = #{userId} AND rno = #{rno}
    </select>

    <insert id="addBoardView" keyProperty="bvno">
        <selectKey keyProperty="bvno" resultType="int" order="BEFORE">
            SELECT COALESCE(MAX(bvno), 0) + 1 AS bvno
            FROM   bbs_boardview
        </selectKey>

        INSERT INTO bbs_boardview(bvno, bno, userId, mem_yn )
        VALUES ( #{bvno}, #{bno}, #{userId}, #{mem_yn})
    </insert>

    <select id="getBoardView" resultType="int">
        SELECT COALESCE(MAX(bno), 0) AS bno
        FROM   bbs_boardview
        WHERE userId = #{userId} AND bno = #{bno}
    </select>
</mapper>
```

root-context에서 dataSource와 Tx 작성

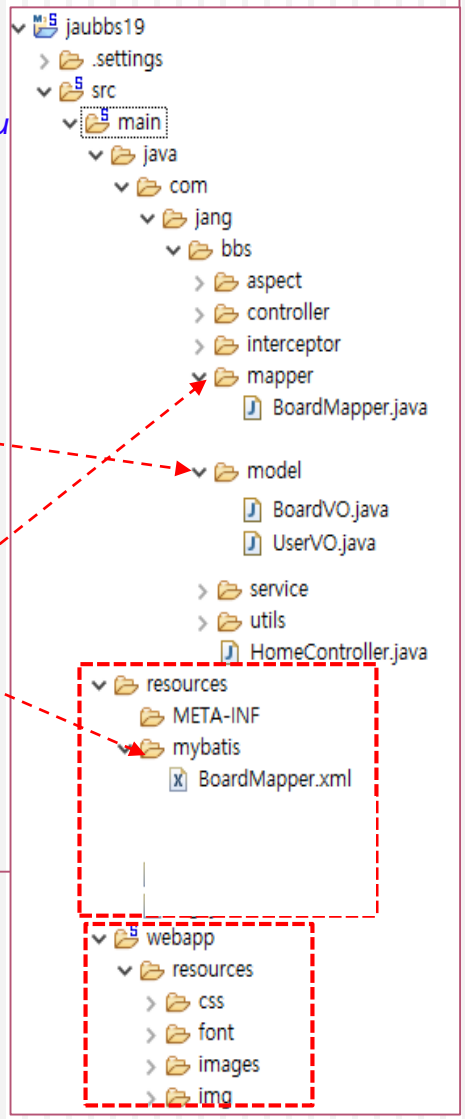
```
<!-- Data Source  -->
<bean id="dataSource" class="org.apache.commons.dbcp2.BasicDataSource" >
    <property name="driverClassName">
        <value>oracle.jdbc.driver.OracleDriver</value>
    </property>
    <property name="url">
        <value>jdbc:oracle:thin:@168.126.146.45:1521:orcl</value>
    </property>
    <property name="username">
        <value>jjin</value>
    </property>
    <property name="password">
        <value>jjinpang</value>
    </property>
    <property name="maxTotal">
        <value>2</value>
    </property>
    <property name="maxIdle">
        <value>5</value>
    </property>
    <property name="maxWaitMillis">
        <value>10000</value>
    </property>
</bean>

<!-- enable transaction demarcation with annotations -->
<tx:annotation-driven />

<!-- transaction manager, use JtaTransactionManager for global tx -->
<bean id="transactionManager" class="org.springframework.jdbc.datasource.DataSourceT
    <property name="dataSource" ref="dataSource" />
</bean>
```

```
<beans xmlns="http://www.springframework.org/schema/b
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-inst
    xmlns:aop="http://www.springframework.org/schema
    xmlns:tx="http://www.springframework.org/schema/
    xmlns:jdbc="http://www.springframework.org/schem
    xmlns:context="http://www.springframework.org/sc
    xsi:schemaLocation="
    http://www.springframework.org/schema/context ht
    http://www.springframework.org/schema/beans http
    http://www.springframework.org/schema/jdbc http:
    http://www.springframework.org/schema/tx http://
    http://www.springframework.org/schema/aop http:/
```



sqlSessionFactory 객체 생성, Mapper Scan

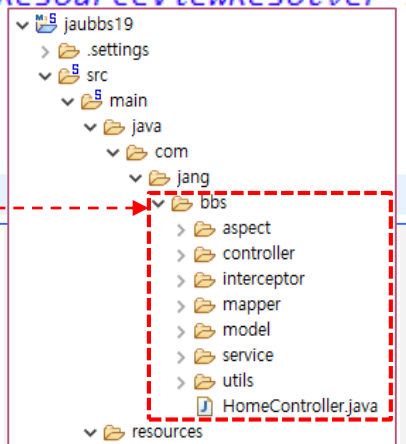
```
<!-- enable transaction demarcation with annotations -->
<tx:annotation-driven />

<!-- transaction manager, use JtaTransactionManager for global tx -->
<bean id="transactionManager" class="org.springframework.jdbc.datasource.DataSou
        <property name="dataSource" ref="dataSource" />
</bean>

<!-- define the SqlSessionFactory -->
<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
     <property name="dataSource" ref="dataSource" />
     <property name="typeAliasesPackage" value="com.jang.bbs.model" />
     <property name="mapperLocations">
         <list>
             <value>classpath*:mybatis/*Mapper.xml</value>
         </list>
     </property>
</bean>

<!-- scan for mappers and let them be autowired -->
<bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">
     <property name="basePackage" value="com.jang.bbs.mapper" />
</bean>
```



*Mapper.java 파일 내용과 *Mapper.xml 파일 내용이 일치 하지않으면 error 발생
(메서드명-id 불일치, id 중복, 내용 없음, SQL 파라메터 표기 실수 등)

Servlet-Context 확인

```
<?xml version="1.0" encoding="UTF-8"?>
<beans:beans xmlns="http://www.springframework.org/schema/mvc"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:beans="http://www.springframework.org/schema/beans"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:mvc="http://www.springframework.org/schema/mvc"
    xsi:schemaLocation="http://www.springframework.org/schema/mvc http://www.springframework
        http://www.springframework.org/schema/beans http://www.springframework.org/schema/be
        http://www.springframework.org/schema/context http://www.springframework.org/schema/
        http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/

    <!-- DispatcherServlet Context: defines this servlet's request-processing infrastructure

    <!-- Enables the Spring MVC @Controller programming model -->
    <annotation-driven />

    <!-- Handles HTTP GET requests for /resources/** by efficiently serving up static resour
    <resources mapping="${webappRoot}/resources/**" location="/resources/" />

    <!-- Resolves views selected for rendering by @Controllers to .jsp resources in the /WEB
    <beans:bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <beans:property name="prefix" value="/WEB-INF/views/" />
        <beans:property name="suffix" value=".jsp" />
    </beans:bean>

    <context:component-scan base-package="com.jang.bbs" />
```



Service 클래스 Paging 처리

스프링 게시판 로그아웃 제목 검색

글번호	제목	작성자	댓글수	조회수	추천수	작성일
12	test10	test10	0	8	0	2015-08-24
11	test9	test9	0	14	0	2015-08-24
10	test8	test8	0	10	2	2015-08-24
9	test7	test7	0	5	0	2015-08-24
8	test6	1212112	0	1	0	2015-08-24
7	test5	test5	0	1	0	2015-08-24
6	test4	ttest4	0	0	0	2015-08-24
5	test3	test3	0	0	0	2015-08-24
4	test2	test2	0	1	0	2015-08-24
3	test	test	0	0	0	2015-08-24

. **1** . 2 . <다음>

목록 쓰기

List 페이지 나누기

게시판

endRow

pageSize(10)

startRow

번호	글 제목	작성자	작성일	조회
36	잘될것이다.	잘잘이	2011-10-12	1
35	잘될것이다.	잘잘이	2011-10-12	2
34	321321312	한작심	2011-10-12	2
33	321321312	한작심	2011-10-12	3
32	121212	33333	2011-10-12	1
31	121212	33333	2011-10-12	1
30	23123	23213	2011-10-12	0
29	23123	23213	2011-10-12	0
28	qweqwee	22313	2011-10-12	0
27	3333333333333333333	3333333333	2011-10-12	2

[글쓰기]

[이전] [11][12][13] [14][15] [16] [7][18][19][20] [다음]

totalPage

=totalRow/pageSize+1

=223행/10+1= 23페이지

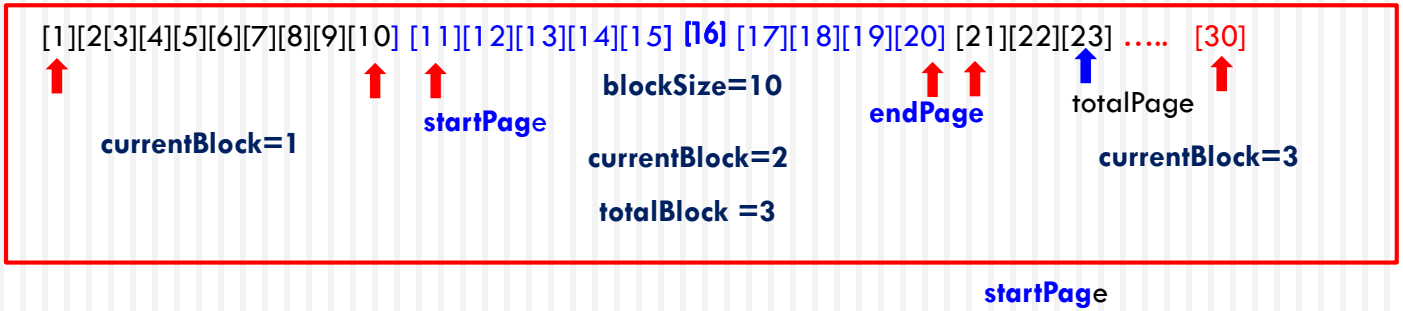
[1][2[3][4][5][6][7][8][9][10] [11][12][13][14][15] [16] [17][18][19][20] [21][22][23] ..... [30]

[이전] startPage ← blockSize=10 → endPage [다음]

currentBlock=2

현재 페이지에서 블록, 시작, 끝행 연산

- list.do?page=16 :현재 페이지 16
- pageSize=10, blockSize=10인 경우
- currentPage = page
- startRow = (currentPage - 1) * pageSize + 1; // 시작 행번호 16 page 경우= (16-1)*10 +1 = 151
- endRow = currentPage * pageSize; // 끝 행번호계산 16 page 경우= 16*10 = 160
- currentBlock = currentPage/blockSize + 1; //현재블록 현재 페이지16 경우: 16/10+1= 2 블록
- startPage = (currentBlock-1) * blockSize + 1; //현재 블럭의 시작 페이지 (2블록-1)*10+1 = 11 page
- endPage = currentBlock * blockSize; //현재 블럭의 끝 페이지 (2블록)*10 = 20 page
- totalPage = totalRow/pageSize+1 : //전체행을 db에서 읽어와 전체 페이지 계산
- totalBlock = totalPage/blockSize + 1; //전체 블럭 수 3 블록=23페이지/10+1



부질의 와 엘리어스(alius)

```
<select id="getBoardList"  parameterType="SearchVO" resultType="Board" >

  SELECT  rnum, bno, title, writerid, replycnt, viewcnt, to_char(regdate, 'YYYY-MM-DD HH24:MI')
         regdate, recommendcnt
  FROM  (  SELECT ROWNUM AS rnum, a.*
           FROM   ( SELECT * FROM   bbs_board  WHERE  del_yn='n' order by bno desc  ) a
         )
  WHERE rnum  between #{startRow} and #{endRow}

 </select>
```

BNO	TITLE	CONTENT
27	cvcxv	cxvcxv
26	page14	page
25	page13	page
24	page12	page
23	page11	page
22	page10	page
21	page9	page
20	page8	page
19	page7	page
18	page6	page
17	page5	page
16	page4	page
15	page3	page
14	page2	page
13	g1121	212112
12	fgfgdf	gdfgdf
11	이제 검색 기능만 다하면 된다.	ㄴㅇㄹㄴㅇ
10	첨부 파일 크기	첨부파일이 100 M 5
6	Test67788	qwqwqwqwqw
5	Test 5	dfsdfsf
2	Test2	sdsadsd
1	Test1	good luck!

RNUM	BNO	TITLE	CONTENT
1	27	cvcxv	cxvcxv
2	26	page14	page
3	25	page13	page
4	24	page12	page
5	23	page11	page
6	22	page10	page
7	21	page9	page
8	20	page8	page
9	19	page7	page
10	18	page6	page
11	17	page5	page
12	16	page4	page
13	15	page3	page
14	14	page2	page
15	13	g1121	212112
16	12	fgfgdf	gdfgdf
17	11	이제 검색 기능만 다하면 된다.	ㄴㅇㄹㄴㅇ
18	10	첨부 파일 크기	첨부파일이 100
19	6	Test67788	qwqwqwqwqw
20	5	Test 5	dfsdfsf
21	2	Test2	sdsadsd
22	1	Test1	good luck!

#{startRow} =3
#{endRow}=12

RNUM	BNO	TITLE	WRITERID
3	25	page13	TestId
4	24	page12	TestId
5	23	page11	TestId
6	22	page10	TestId
7	21	page9	TestId
8	20	page8	TestId
9	19	page7	TestId
10	18	page6	TestId
11	17	page5	TestId
12	16	page4	TestId

PageHelper 클래스 구성

```java
package com.jang.bbs.utils;

public class PageHelper {

    private int pageSize = 10; // 한 페이지당 보이는 게시물 수
    private int blockSize = 10; //한 블럭당 보이는 페이지 수

    public int getPageSize() {
        return pageSize;
    }

    public int getBlockSize() {
        return blockSize;
    }

    // 현재 페이지와 전체 레코드수를 입력 받아 페이징 표시 문자열을 반환한다.
    public StringBuffer getPageUrl(int currentPage, int totalRow) {

        StringBuffer pageHtml = new StringBuffer();
        Int startPage = 0;
        Int endPage = 0;

        //표시할 시작페이지와 끝페이지 계산
        startPage = ((currentPage-1) / blockSize) * blockSize + 1;
        endPage = startPage + blockSize - 1;

        //전체 페이지 수와 블럭수 계산
        Int totalPage = totalRow/pageSize + 1;
        int totalBlock = totalPage/blockSize + 1;
        int currentBlock = currentPage/blockSize + 1;
```

Utils package 에 PageHelper 클래스 작성

검색시 필요

```java
public class SearchVO {
    private String type = "";
    private String keyword = "";
    private int   page=1;
    private int   startRow=1;
    private int   endRow=1;
}
```

currentPage, totalRow

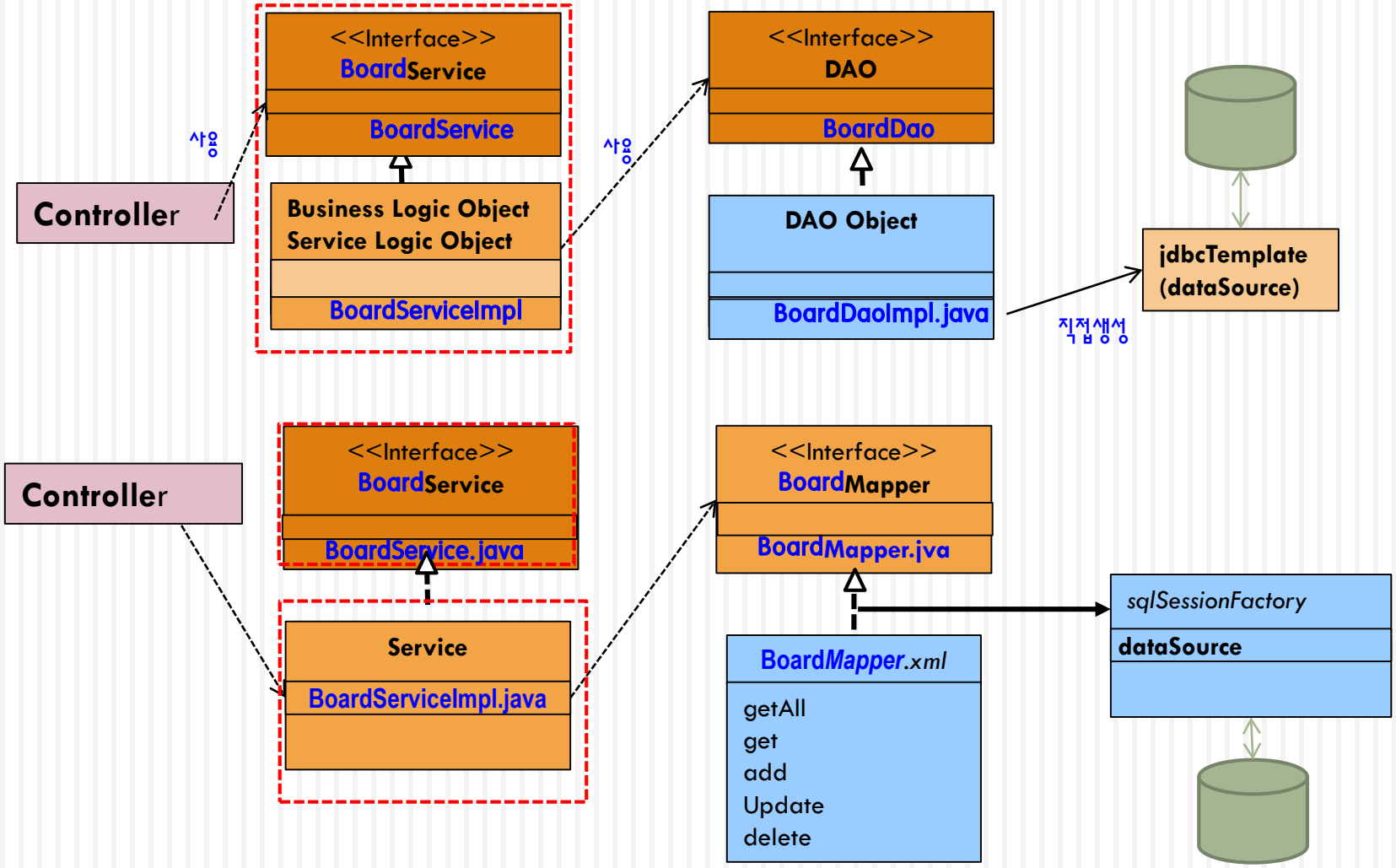
getPageUrl()

.1.2.<다음>

PageHelper 클래스 구성

```
      //표시되는 현재 블럭의 끝 페이지가 전체 마지막 페이지인 경우- 전체 페이지보다 큰 경우 발생시  조정  : 223행/10 + 1 = 23페이지
      if( endPage > (totalRow / pageSize ) ) {
            endPage = (totalRow /pageSize) + 1;
      }
      //현재 블럭이 2 이상인 경우 [이전] 표시
      if(currentBlock > 1){
            pageHtml.append("<span><a href=\"list.do?page=" + (currentPage-1) + "\"><이전></a>  ");
      }else{
            pageHtml.append("<span>");
      }
      //현재 블럭안에 포함될  페이지 표시
      for(int i = startPage ; i <= endPage ; i++) {
            if(i == currentPage){
                  pageHtml.append(". <strong>");
                  pageHtml.append( i );
                  pageHtml.append(" </strong>");
            } else {
              pageHtml.append(". <a href=\"list.do?page=" +i + "\" class=\"page\">" + i + "</a> ");
            }
      }
      //현재 블록 안이 전체 블럭 보다 작으면 [다음] 표시
      if(currentBlock < totalBlock){
           pageHtml.append(".  <a href=\"list.do?page=" + (currentPage+1) + "\"><다음></a></span>");
      }else{
            pageHtml.append(".</span>");
      }

      return pageHtml;
   }
}
```

Service (Business)클래스



BoardService Interface 작성

```java
package com.jang.bbs.service;

import java.util.List;

public interface BoardService {

    StringBuffer getPageUrl (SearchVO searchVO );

    List<BoardVO> getBoardList(SearchVO searchVO); //글 목록조회
    BoardVO getArticle(int bno);        //글 조회
    int writeArticle(BoardVO board);    //글저장
    int updateArticle(BoardVO board);   //글수정
    void deleteArticle(int bno);        //글삭제

    //int getTotalRow(SearchVO searchVO); //Controller에서는 사용되지 않음

    int incrementViewCnt(int bno);  //조회수 증가
    int incrementGoodCnt(int bno);  //좋아요 증가
    int incrementReplyCnt(int bno); //댓글 수 증가

    List<ReplyVO> getReplyList(int bno);  //댓글 목록조회
    ReplyVO getReply(int rno);  // 댓글조회

    int writeReply(ReplyVO reply); //댓글저장
    int updateReply(ReplyVO reply); //댓글수정
    void deleteReply(int rno);      //댓글삭제
    void deleteReplyBybno(int bno); //원글에 소속된 댓글 전체 삭제

    int incReplyGoodCnt(int rno); //댓글 좋아요 증가

    List<AttFileVO> getFileList( int bno); //첨부파일 목록 조회
    String getFileName(int fno);  //첨부파일 이름 조회
    int insertFile(AttFileVO file); //첨부파일 저장

    void deleteFile(int fno);  //첨부파일 삭제
    void deleteFileBybno(int bno); //원글에 소속된 첨부파일 전체 삭제

    int addBoardLike(BoardLikeVO boardLike); //id-특정글 좋아요 기록 저장
    int getBoardLike(BoardLikeVO boardLike); //id-특정글 좋아요 기록 조회

    int addReplyLike(ReplyLikeVO replyLike); //id-특정댓글 좋아요 기록 저장
    int getReplyLike(ReplyLikeVO replyLike); //id-특정댓글 좋아요 기록 조회

    int addBoardView(BoardViewVO boardView); //id-특정글  조회이력  기록
    int getBoardView(BoardViewVO boardView); //id-특정글  조회이력  읽기
```

BoardServiceImpl 클래스 작성

```java
package com.jang.bbs.service;

import java.util.List;


@Service(value = "boardService")
public class BoardServiceImpl implements BoardService {

    @Autowired //@Resource(name = "boardMapper")
    private BoardMapper boardMapper;

    PageHelper pageHelper = new PageHelper();

    @Override
    public StringBuffer getPageUrl(SearchVO searchVO) {

        int totalRow =  boardMapper.getTotalRow(searchVO);

        return pageHelper.getPageUrl(searchVO.getPage(), totalRow);
    }

    @Override
    public List<BoardVO> getBoardList(SearchVO searchVO) {
        int currentPage = searchVO.getPage();

        int startRow = (currentPage - 1) * this.pageHelper.getPageSize()+1 ;
        int endRow   = currentPage *  this.pageHelper.getPageSize();

        searchVO.setStartRow(startRow);
        searchVO.setEndRow(endRow);

        return this.boardMapper.getBoardList(searchVO);
    }
```

검색시 필요

```java
public class SearchVO {
    private String type = "";
    private String keyword = "";
    private int    page=1;
    private int    startRow=1;
    private int    endRow=1;
}
```

currentPage, totalRow

getPageUrl()

.1.2.<다음>

BoardService 클래스 작성

```java
@Override
public BoardVO getArticle(int bno) {
     return this.boardMapper.getArticle(bno);
}

@Override
public int writeArticle(BoardVO board) {
    return this.boardMapper.writeArticle(board);
}

@Override
public int updateArticle(BoardVO board) {
    return this.boardMapper.updateArticle(board);
}

@Override
public void deleteArticle(int bno) {
    this.boardMapper.deleteArticle(bno);
}

@Override
public int incrementViewCnt(int bno) {
    return this.boardMapper.incrementViewCnt(bno);
}

@Override
public int incrementGoodCnt(int bno) {
    return this.boardMapper.incrementGoodCnt(bno);
}

@Override
public int incrementReplyCnt(int bno) {
    return this.boardMapper.incrementReplyCnt(bno);
}
```

Mybatis 반환값 :

Select - Select문에 해당하는결과

insert - 1 (여러개일경우도 1)

Update - Update된 행의 개수 반환 (없다면 0)

delete - Delete된 행의개수 (없다면 0)

```java
@Override
public List<ReplyVO> getReplyList(int bno) {
    return this.boardMapper.getReplyList(bno);
}

@Override
public ReplyVO getReply(int rno) {
    return this.boardMapper.getReply(rno);
}

@Override
public int writeReply(ReplyVO reply) {
    int bno = reply.getBno();
    this.boardMapper.incrementReplyCnt(bno);
    return this.boardMapper.writeReply(reply);
}

@Override
public int updateReply(ReplyVO reply) {
    return this.boardMapper.updateReply(reply);
}

@Override
public void deleteReply(int rno) {
    this.boardMapper.deleteReply(rno);
}
```

BoardService 클래스 작성

```java
@Override
public void deleteReplyBybno(int bno) {
    this.boardMapper.deleteReplyBybno(bno);
}

@Override
public int incReplyGoodCnt(int rno) {
    return this.boardMapper.incReplyGoodCnt(rno);
}

@Override
public List<AttFileVO> getFileList(int bno) {
    return this.boardMapper.getFileList(bno);
}

@Override
public String getFileName(int fno) {
    return this.boardMapper.getFileName(fno);
}

@Override
public int insertFile(AttFileVO file) {
    return this.boardMapper.insertFile(file);
}

@Override
public void deleteFile(int fno) {
    this.boardMapper.deleteFile(fno);
}

@Override
public void deleteFileBybno(int bno) {
    this.boardMapper.deleteFileBybno(bno);
}
```

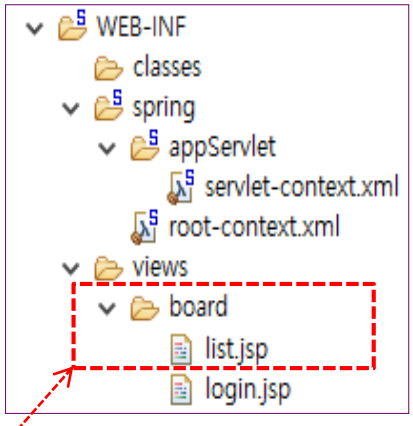
```java
    @Override
    public int addBoardLike(BoardLikeVO boardLike) {
        return this.boardMapper.addBoardLike(boardLike);
    }

    @Override
    public int getBoardLike(BoardLikeVO boardLike) {
        return this.boardMapper.getBoardLike(boardLike);
    }

    @Override
    public int addReplyLike(ReplyLikeVO replyLike) {
        return this.boardMapper.addReplyLike(replyLike);
    }

    @Override
    public int getReplyLike(ReplyLikeVO replyLike) {
        return this.boardMapper.getReplyLike(replyLike);
    }

    @Override
    public int addBoardView(BoardViewVO boardView) {
        return this.boardMapper.addBoardView(boardView);
    }

    @Override
    public int getBoardView(BoardViewVO boardView) {
        return this.boardMapper.getBoardView(boardView);
    }

}
```

BoardController 작성 (list.do)

```java
package com.jang.bbs.controller;

import java.io.File;

@Controller
@RequestMapping("/board")
public class BoardController {

    @Autowired //@Resource(name = "boardService")  의존관계 주입
    private BoardService boardService; //의존관계 설정

    // file upload path
    private String uploadPath = "C:\\upload\\";

    @RequestMapping(value = "/list", method = RequestMethod.GET)
    public String listPage(@ModelAttribute("searchVO") SearchVO searchVO, Model model, HttpSession session) throws Exception {

        //session.setAttribute("userId", "jjin33");   // login 개발 완료후 삭제
        session.setAttribute("userName", "Tester");  // login 개발 완료후 삭제

        List<BoardVO> blist = boardService.getBoardList(searchVO) ;
        model.addAttribute("boardList", blist);

        StringBuffer pageUrl = boardService.getPageUrl(searchVO) ;
        model.addAttribute("pageHttp",  pageUrl);

        model.addAttribute("userId",  session.getAttribute("userId"));
        model.addAttribute("searchVO",  searchVO);

        return "board/list";
    }
```

•/board/list.do?page=16

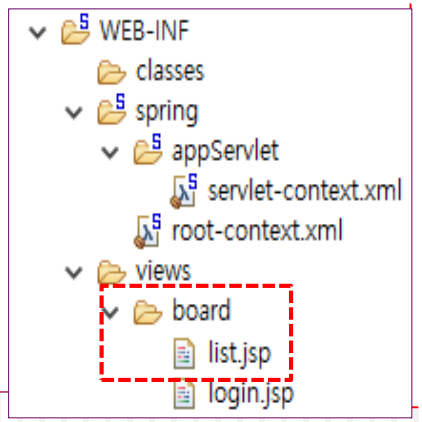
```java
public class SearchVO {
    private String type = "";
    private String keyword = "";
    private int   page=1;
    private int   startRow=1;
    private int   endRow=1;
}
```

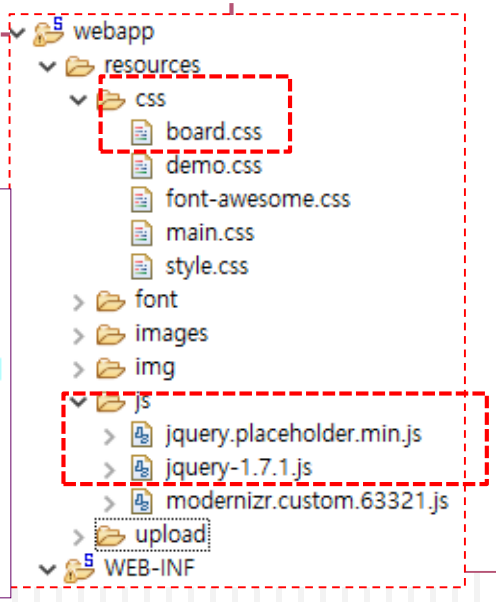


list.jsp(script 부분)

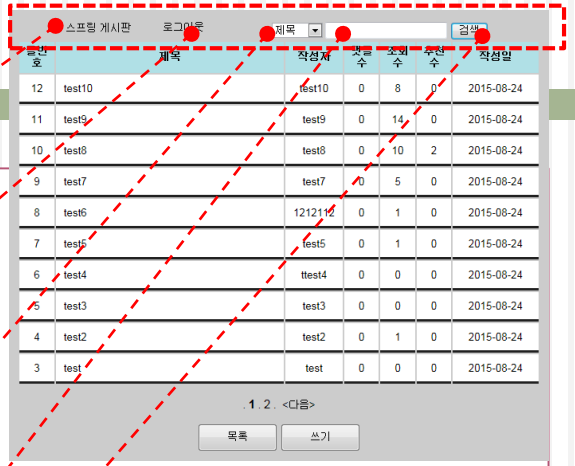
```
<%@ page language="java" contentType="text/html; charset=UTF-8"  pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>게시판 글 목록</title>

<link href="${pageContext.request.contextPath}/resources/css/board.css" rel="stylesheet" type="text/css" />

<script type="text/javascript" src="${pageContext.request.contextPath}/resources/js/jquery-1.7.1.js"></script>

<script type="text/javascript">

function selectedOptionCheck(){
    $("#type > option[value=<%=request.getParameter("type")%>]").attr("selected", "true");
}
                                  [value=${param.type}]

function moveAction(where){
      switch (where) {
            case 1:
            location.href = "write.do";
            break;

            case 2:
            location.href = "list.do";
            break;
      }
}

</script>
</head>
```

jquery를 이용하여 select option 선택값 변경내용 현재화면에 표기





list.jsp(검색 부분)

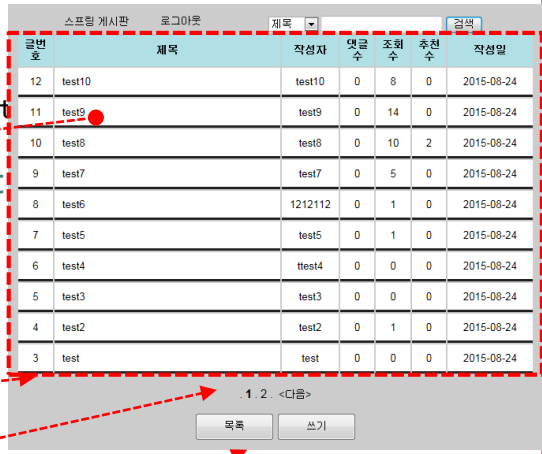



```
<body onload="selectedOptionCheck()">
    <div class="wrapper">
        <div class="navBar">
        <ul>
            <li><a href="list.do">스프링 게시판</a></li>
            <li><a href="../member/logout.do">로그아웃</a></li>
        </ul>
        <form action="list.do"  modelAttribute="searchVO" method="get">
            <select id="type" name="type">
                <option value="title">제목</option>
                <option value="content">내용</option>
                <option value="writerId">작성자</option>
            </select>
            <input type="text" id="keyword" name="keyword"
            value="<%if(request.getParameter("keyword") != null)
                        { out.print(request.getParameter("keyword")); }
                    else { out.print(""); }%>" />
            <input type="submit" value="검색" />
        </form>
        </div>
```

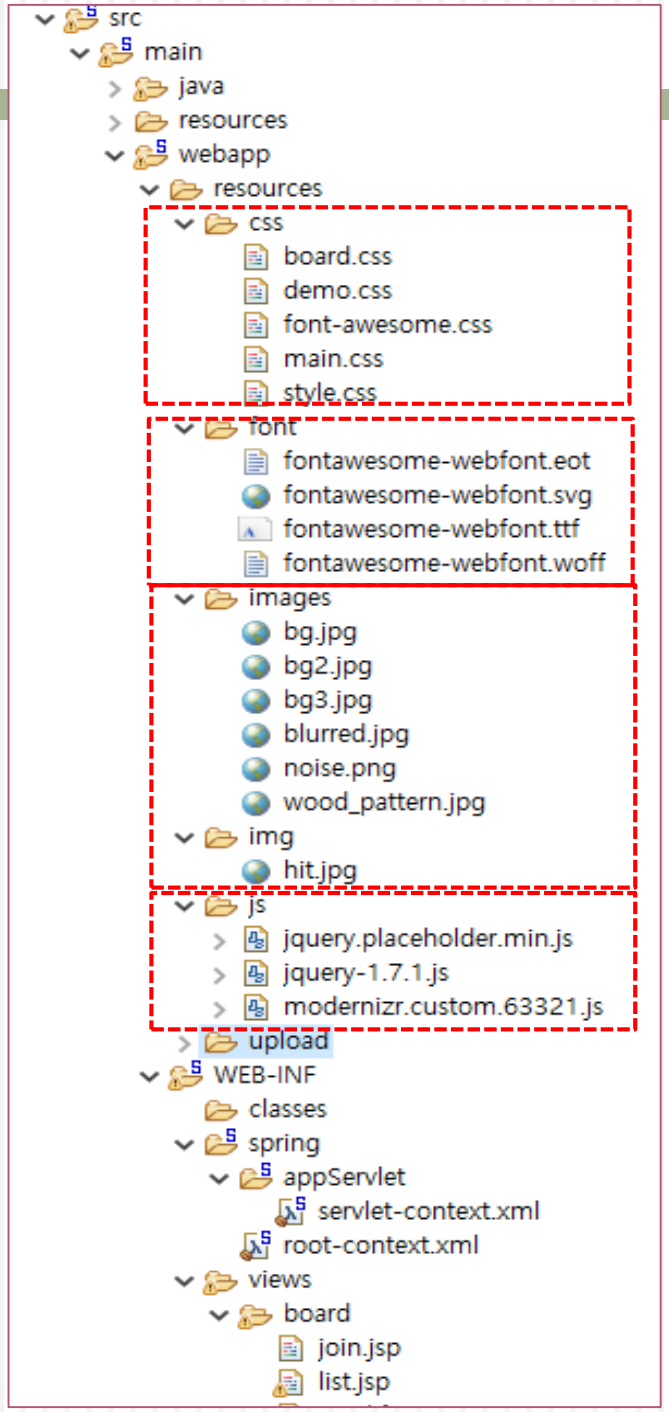
검색버튼에서 list를 표시할 경우 현재 선택된 내용을 보여준다.

list.jsp (테이블 표시부분)

```
<table border="0" class="boardTable">
<thead>
<tr>
   <th>글번호</th><th>제목</th><th>작성자</th><th>댓글수</th><th>조회수</th><th>추천수</th><th>작성일</th>
</tr>
</thead>
<tbody>
<c:forEach var="board" items="${boardList}">
<tr>
     <td class="idx">${board.bno}</td>
     <td align="left" class="subject">
     <c:if test="${board.replycnt >= 10}"><img src="${pageContext.request.context
     /></c:if>
     <a href="view.do?bno=${board.bno}">${board.title}</a></t
     <td class="writer">  ${board.writerId}</td>
     <td class="comment">${board.replycnt}</td>
     <td class="hitcount">${board.viewcnt}</td>
     <td class="recommendcount">${board.recommendcnt}</td>
     <td class="writeDate">${board.regDate}</td>
</tr>
</c:forEach>
</tbody>
</table>
<br />
${pageHttp}
<br /><br />
<input type="button" value="목록" class="writeBt" onclick="moveAction(2)"/>
<input type="button" value="쓰기" class="writeBt" onclick="moveAction(1)"/>
</div>
```



Resource 설치


src
main
java
resources
webapp
resources
css
board.css
demo.css
font-awesome.css
main.css
style.css
font
fontawesome-webfont.eot
fontawesome-webfont.svg
fontawesome-webfont.ttf
fontawesome-webfont.woff
images
bg.jpg
bg2.jpg
bg3.jpg
blurred.jpg
noise.png
wood_pattern.jpg
img
hit.jpg
js
jquery.placeholder.min.js
jquery-1.7.1.js
modernizr.custom.63321.js
upload
WEB-INF
classes
spring
appServlet
servlet-context.xml
root-context.xml
views
board
join.jsp
list.jsp

List 중간 결과 실행(검색 구현 안됨)

http://localhost:8080/bbs/board/list.do

스프링 게시판 로그아웃 제목 검색

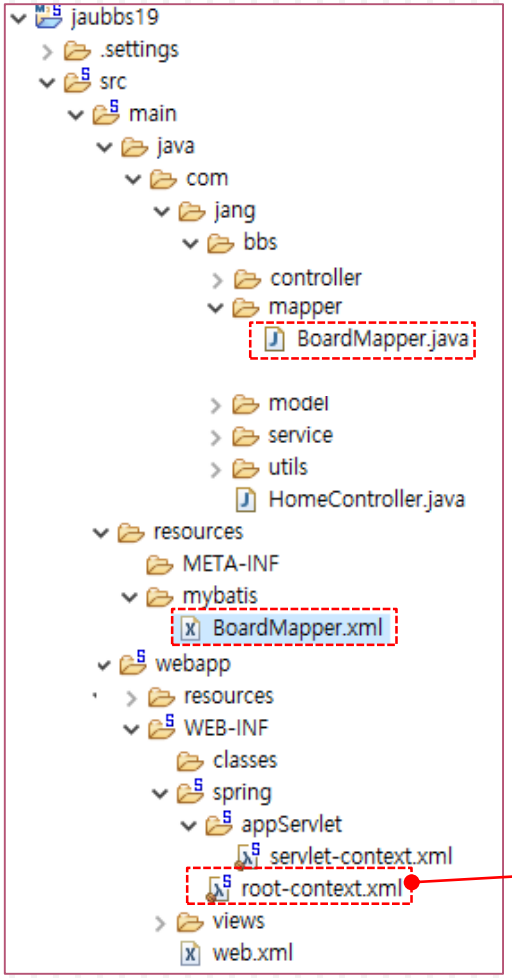
글번호	제목	작성자	댓글수	조회수	추천수	작성일
28	121212324242	TestId	6	1	0	2018-10-22 10:17
27	cvcxv	TestId	0	3	0	2018-10-19 13:35
26	page14	TestId	2	3	0	2018-10-19 13:34
25	page13	TestId	1	1	0	2018-10-19 13:34
24	page12	TestId	0	0	0	2018-10-19 13:33
23	page11	TestId	1	3	0	2018-10-19 13:33
22	page10	TestId	0	1	0	2018-10-19 13:33
20	page8	TestId	0	0	0	2018-10-19 13:33
19	page7	TestId	0	2	0	2018-10-19 13:33
18	page6	TestId	0	0	0	2018-10-19 13:32

. 1 . 2 . 3 .

목록 쓰기

Mybatis 기본 동작

Root-context.xml 에서 sqlSessionFactory 객체가 생성 될 때
*Mapper.java 파일과 *Mapper.xml 파일을 대응시켜 객체를 생성한다.



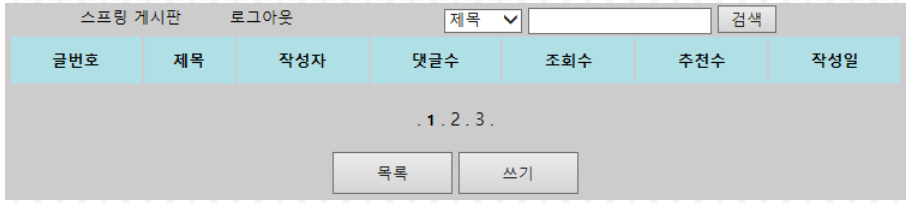

```xml
<?xml version="1.0" encoding="UTF-8"?>

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

<mapper namespace="com.jang.bbs.mapper.BoardMapper">

    <select id="getArticle" resultType="BoardVO">
        SELECT  bno , title, content, writerId, to_char(regDate, '
         FROM   bbs_board
         WHERE  bno = #{bno}
    </select>

    <insert id="writeArticle" keyProperty="bno">
        <selectKey keyProperty="bno" resultType="int" order="BEFO
            SELECT COALESCE(MAX(bno), 0) + 1 AS bno   FROM   bbs_
        </selectKey>

        INSERT INTO bbs_board (bno, title, content, writerId) VAL
    </insert>
```

```java
@Mapper//@Repository(value = "boardMapper")
public interface BoardMapper {

    List<BoardVO> getBoardList(SearchVO searchVO);

    BoardVO getArticle(int bno);          //글 조회
    int writeArticle(BoardVO board);      //글저장
    int updateArticle(BoardVO board);     //글수정
    void deleteArticle(int bno);          //글삭제

    int getTotalRow(SearchVO searchVO); //전체글 수 조

    int incrementViewCnt(int bno);   //조회수 증가
    int incrementGoodCnt(int bno);   //좋아요 증가
    int incrementReplyCnt(int bno); //댓글 수 증가
```

```xml
<!-- define the SqlSessionFactory -->
<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
    <property name="dataSource" ref="dataSource" />
    <property name="typeAliasesPackage" value="com.jang.bbs.model" />
    <property name="mapperLocations">
        <list>
            <value>classpath*:mybatis/*Mapper.xml</value>
        </list>
    </property>
</bean>

<!-- scan for mappers and let them be autowired -->
<bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">
    <property name="basePackage" value="com.jang.bbs.mapper" />
</bean>
```

Error 수정

- @Mapper가 없을 때
  - Caused by: java.lang.ClassNotFoundException: javax.xml.bind.ValidationException
- DTO 클래스 명이 일치 하지않을 때 (입력, 출력)
  - Caused by: org.apache.ibatis.type.TypeException: Could not resolve type alias ' BoardVO2 ' . Cause: java.lang.ClassNotFoundException: Cannot find class: BoardVO2
  - Caused by: org.apache.ibatis.builder.BuilderException: Error resolving class. Cause: org.apache.ibatis.type.TypeException: Could not resolve type alias ' SearchVO2 ' . Cause: java.lang.ClassNotFoundException: Cannot find class: SearchVO2
- Sql 명령어의 필드명이 테이블과 일치하지 않을 때
  - INFO : org.springframework.beans.factory.xml.XmlBeanDefinitionReader - Loading XML bean definitions from class path resource [org/springframework/jdbc/support/sql-error-codes.xml] exception!
- 기타 :



Error message

```
<select id="getBoardList"  parameterType="SearchVO" resultType="Board" >
```

클래스명이 틀린 경우

```
org.springframework.beans.factory.BeanCreationException: Error creating bean with name
'sqlSessionFactory' defined in ServletContext resource [/WEB-INF/spring/root-context.xml]: Invocation
of init method failed; nested exception is org.springframework.core.NestedIOException: Failed to
parse mapping resource: 'file [C:\eclipse\photon-
workspace\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\jaubbs18\WEB-
INF\classes\mybatis\BoardMapper.xml]'; nested exception is org.apache.ibatis.builder.BuilderException:
Error parsing Mapper XML. The XML location is 'file [C:\eclipse\photon-
workspace\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\jaubbs18\WEB-
INF\classes\mybatis\BoardMapper.xml]'. Cause: org.apache.ibatis.builder.BuilderException: Error
resolving class. Cause: org.apache.ibatis.type.TypeException: Could not resolve type alias 'Board'.
Cause: java.lang.ClassNotFoundException: Cannot find class: Board
```

```
<!-- define the SqlSessionFactory -->
<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
    <property name="dataSource" ref="dataSource" />
    <property name="typeAliasesPackage" value="com.jang.bbs.model" />
    <property name="mapperLocations">
        <list>
            <value>classpath*:mybatis/*Mapper.xml</value>
        </list>
    </property>
</bean>
```

*Mapper.java 파일 내용과 *Mapper.xml 파일 내용이 일치 (메서드명-id 불일치 , id 중복, 내용 없음, SQL 파라메터 표기 실수 등) 하지않으면 error 발생

```
org.apache.ibatis.binding.BindingException: Invalid bound statement (not found):
com.jang.bbs.mapper.BoardMapper.getBoardList
org.apache.ibatis.binding.MapperMethod$SqlCommand.<init>(MapperMethod.java:227)
org.apache.ibatis.binding.MapperMethod.<init>(MapperMethod.java:49)
org.apache.ibatis.binding.MapperProxy.cachedMapperMethod(MapperProxy.java:65)
org.apache.ibatis.binding.MapperProxy.invoke(MapperProxy.java:58) com.sun.proxy.$Proxy33.getBoardList(Unknown Source)
com.jang.bbs.service.BoardService.getBoardList(BoardService.java:44)
com.jang.bbs.controller.BoardController.listPage(BoardController.java:48)
```

Error message

Type Status Report

Message /bbs/WEB-INF/views/joinForm.jsp

Description The origin server did not find a current representation for the target resource or is not willing to disclose that one exists.

org.springframework.beans.NotReadablePropertyException: Invalid property 'id' of bean class [com.jang.bbs.model.UserVO]: Bean property 'id' is not readable or has an invalid getter method: Does the return type of the getter match the parameter type of the setter?

```
org.apache.jasper.JasperException: An exception occurred processing [/WEB-INF/views/member/joinForm.jsp] at line [93]

90: <TR>
91:     <TD bgcolor='cccccc' width='100' align='center'><font size='2'>아 이 디</font></TD>
92:     <TD bgcolor='cccccc'>
93:             <form:input path="id" maxLength='20' size='10' value="${userId}" cssClass="userId"  />
94:             <input type='button' OnClick='Check_id()' value='ID 중복검사'>
95:             <span class="fieldError"><form:errors path="userId" /></span><span class="fieldError"> *${message}</span>
96:         <input type="hidden" name="re_id" value="${reDiv}" >
```

메서드명-id 불일치 , id 중복, 내용 없음, SQL 파라메터 표기 실수 등

Invalid bound statement (not found): com.jang.bbs.mapper.LoginMapper.getUser] with root cause
org.apache.ibatis.binding.BindingException: Invalid bound statement (not found): com.jang.bbs.mapper.LoginMapper.getUser
at
org.apache.ibatis.binding.MapperMethod$SqlCommand.<init>(MapperMethod.java:227)

검색 기능의 구현

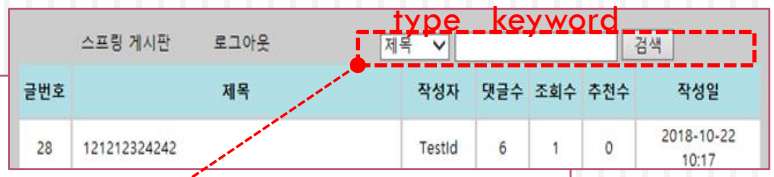
- Type을 선택하고 Keyword 를 입력 후 검색 버튼을 클릭하면 검색 조건에 일치된 내용만 출력되어야 하고 page 계산도 다르게 한다.
  - 검색 버튼의 클릭 여부는 searchVO keyword 항목의 null 여부로 판단한다.
  - getBoardList( ) ➔ getBoardList(searchVO searchVO)
  - getTotalRow( ) ➔ getTotalRow(SearchVO searchVO);
  - SQL XML 파일의 변경 필요

```
public class SearchVO {
    private String type = "";
    private String keyword = "";
    private int   page=1;
    private int   startRow=1;
    private int   endRow=1;
}
```

- list.jsp 에서의 처리

검색 기능의 구현

```
<script type="text/javascript">

    function selectedOptionCheck(){
        $("#type > option[value=${param.type}]").attr("selected", "true");
    }
```




```
<body onload="selectedOptionCheck()">
<div class="wrapper">
    <div class="navBar">
        <ul>
            <li><a href="list.do">스프링 게시판</a></li>
            <li><a href="../member/logout.do">로그아웃</a></li>
        </ul>
        <form action="list.do"  modelAttribute="searchVO" method="get">
            <select id="type" name="type">
                <option value="title">제목</option>
                <option value="content">내용</option>
                <option value="writerId">작성자</option>
            </select>
            <input type="text" id="keyword" name="keyword" value="<%if(request.getParameter("keyword") != null)
            { out.print(request.getParameter("keyword")); } else { out.print(""); }%>" />
            <input type="submit" value="검색" />
        </form>
    </div>
```

1.View

```
public class SearchVO {
    private String type = "";
    private String keyword = "";
    private int   page=1;
    private int   startRow=1;
    private int   endRow=1;
}
```

```
@RequestMapping(value = "/list", method = RequestMethod.GET)
public String listPage(@ModelAttribute("searchVO") SearchVO searchVO, Model

    //session.setAttribute("userId", "TestId");   // login 개발완료 후 삭제
    session.setAttribute("userName", "Tester"); // login 개발 완료 후 삭제

    List<BoardVO> blist = boardService.getBoardList(searchVO) ;
    model.addAttribute("boardList", blist);

    StringBuffer pageUrl = boardService.getPageUrl(searchVO) ;
    model.addAttribute("pageHttp",  pageUrl);

    model.addAttribute("searchVO",  searchVO);

    return "board/list";
}
```

2.Controller

검색 기능의 구현

```java
@Service(value = "boardService")
public class BoardService {

    @Resource(name = "boardMapper") //autowired
    private BoardMapper boardMapper;

    PageHelper pageHelper = new PageHelper();

    public StringBuffer getPageUrl (SearchVO searchVO ) {

        int totalRow =  boardMapper.getTotalRow(searchVO);

        return pageHelper.getPageUrl(searchVO.getPage(), totalRow);
    }

    public List<BoardVO> getBoardList(SearchVO searchVO) {

        int currentPage = searchVO.getPage();

        int startRow = (currentPage - 1) * this.pageHelper.getPageSize()+1 ;
        int endRow   = currentPage *  this.pageHelper.getPageSize();

        searchVO.setStartRow(startRow);
        searchVO.setEndRow(endRow);

        return this.boardMapper.getBoardList(searchVO);
    }
```

```java
public class SearchVO {
    private String type = "";
    private String keyword = "";
    private int   page=1;
    private int   startRow=1;
    private int   endRow=1;
}
```

3. Service * 수정 - error 수정

4. Mapper(java)-수정

```java
@Repository(value = "boardMapper") //@Mapper
public interface BoardMapper {

    List<BoardVO> getBoardList(SearchVO searchVO); //글 목록 조회

    BoardVO getArticle(int bno);       //글조회
    int writeArticle(BoardVO board); //글저장
    int updateArticle(BoardVO board); //글수정
    void deleteArticle(int bno);      //글삭제

    int getTotalRow(SearchVO searchVO); //전체 글수 조회
```

조회

5. Mapper(xml)

제목 | 검색

```
<select id="getBoardList"  parameterType="SearchVO" resultType="Board" >

  SELECT  bno, title, writerid, replycnt, viewcnt, to_char(regdate, 'YYYY-MM-DD HH24:MI')
          regdate, recommendcnt
  FROM  ( SELECT ROWNUM AS rnum, a.*
          FROM ( SELECT * FROM   bbs_board  WHERE  del_yn='n'
                 <if test="keyword != null">
                 <choose>
                    <when test="type == 'title'"> AND title like '%'||#{keyword}||'%' </when>
                    <when test="type == 'writerId'">AND writerId  like '%'||#{keyword}||'%' </when>
                    <when test="type == 'content'"> AND content  like '%'||#{keyword}||'%' </when>
                 </choose>
                 </if>
                 order by bno desc  ) a
        )
  WHERE rnum  between #{startRow} and #{endRow}

 </select>
```

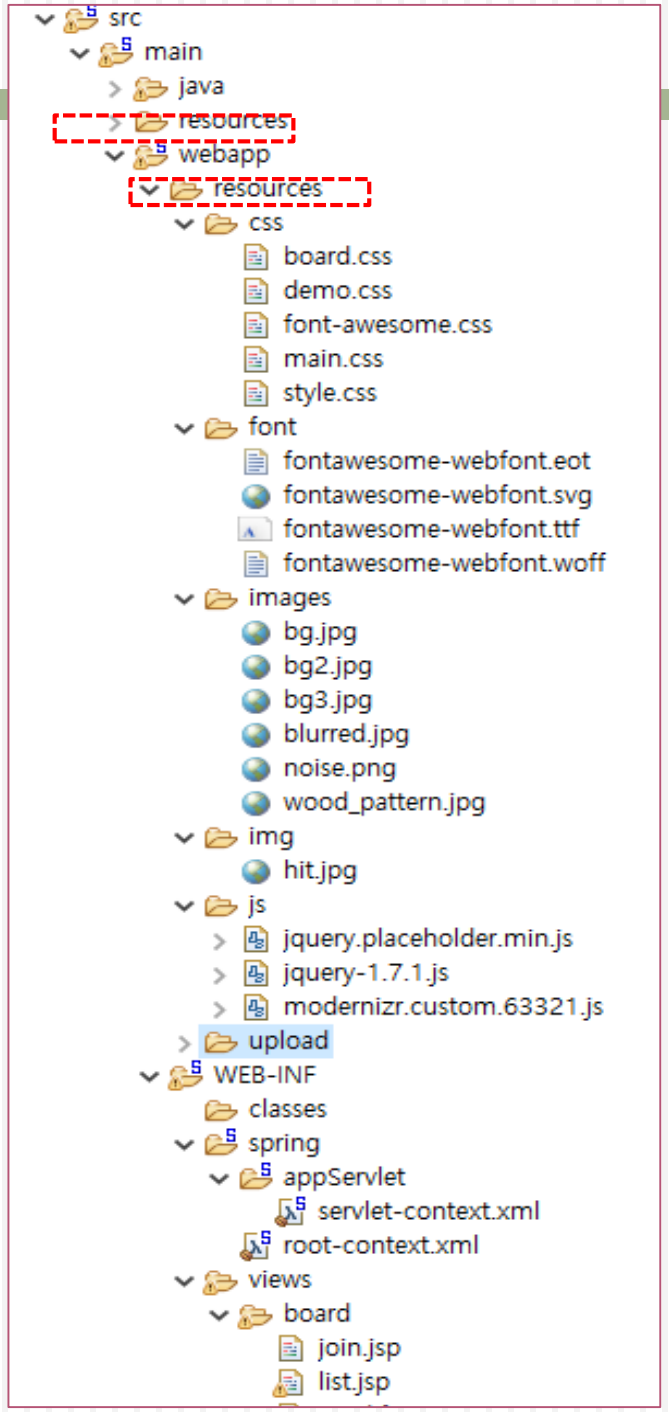
```
@Repository(value = "boardMapper") //@Mapper
public interface BoardMapper {

   List<BoardVO> getBoardList(SearchVO searchVO); //글목록 조회

   BoardVO getArticle(int bno);       //글조회
   int writeArticle(BoardVO board); //글저장
   int updateArticle(BoardVO board); //글수정
   void deleteArticle(int bno);      //글삭제

   int getTotalArticle(SearchVO searchVO); //전체 글수 조회
```

수정

```
<select id="getTotalRow"  parameterType="SearchVO"  resultType="int">
     SELECT count(*) AS totalrow
     FROM  bbs_board
     WHERE del_yn='n'
     <if test="keyword != null">
         <choose>
              <when test="type == 'title'">AND title like '%'||#{keyword}||'%' </when>
              <when test="type == 'writerId'">AND writerId  like '%'||#{keyword}||'%' </when>
              <when test="type == 'content'">AND content  like '%'||#{keyword}||'%' </when>
          </choose>
      </if>
   </select>
```

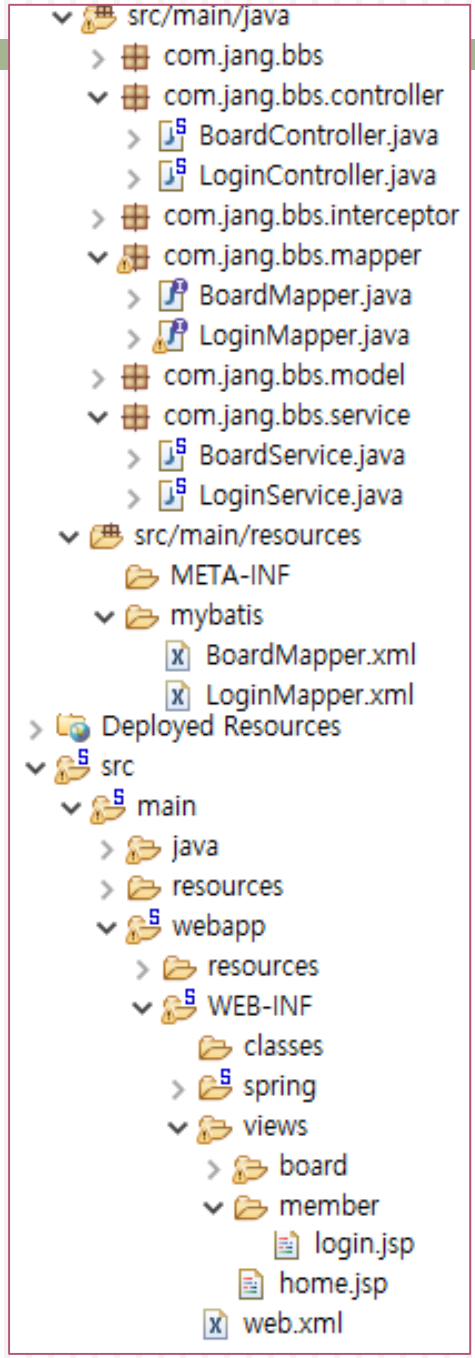
검색 실행

- Webapp/resource 에
  - Css, font, jquery image 설치 확인
- 실행
- 나머지 기능은 개인별로 작성 완성

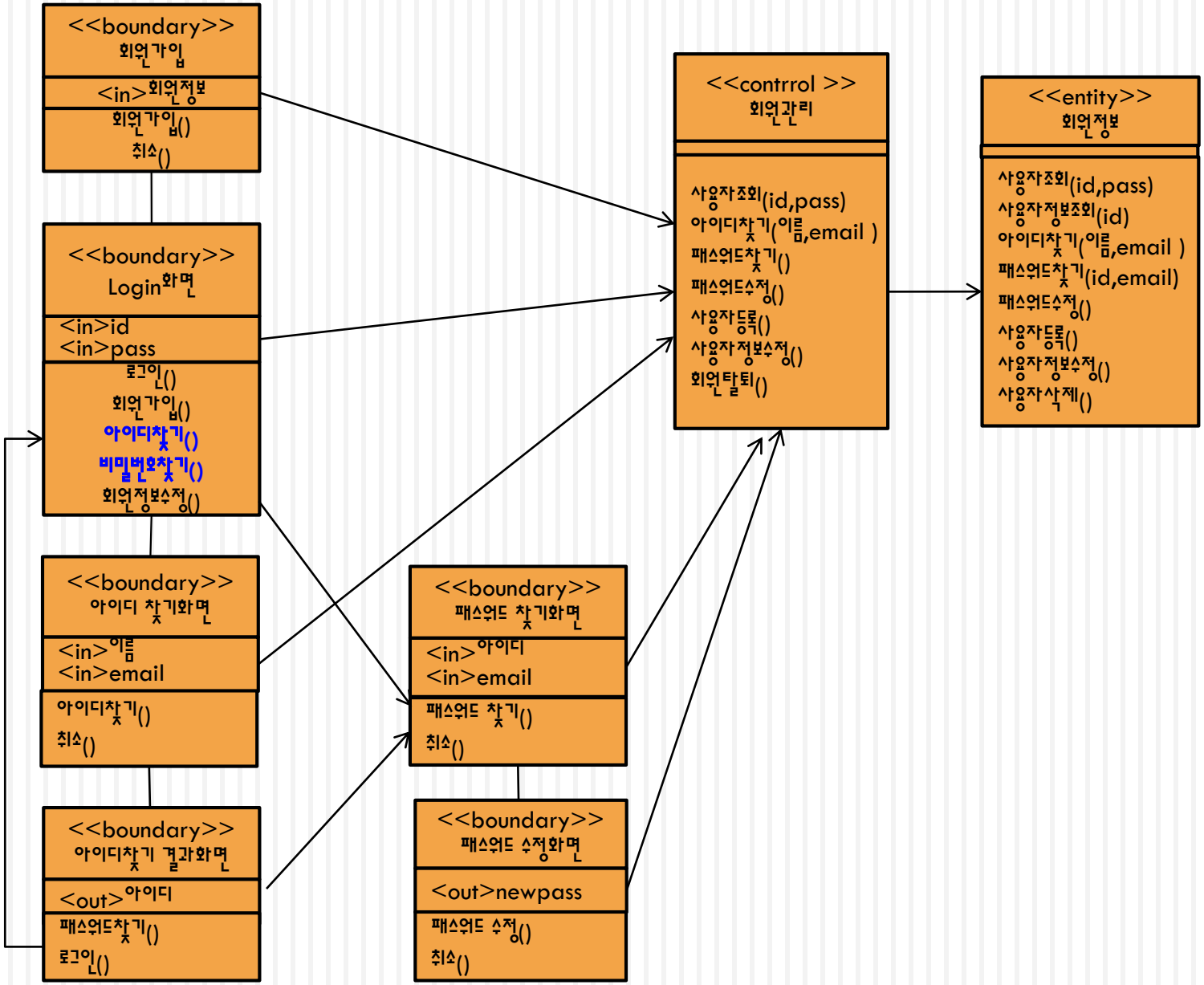


Login 작성 (별도의 Controller 작성)

1. Dao(Mapper) Interface 작성 - LoginMapper.java
2. Mapper.XML 작성 - LoginMapper.XML
3. Service interface 작성 - LoginService.java
4. Service 구현체 작성 - LoginServiceImpl.java
5. Controller 작성 - LoginController.java
6. View 작성 - Login.jsp

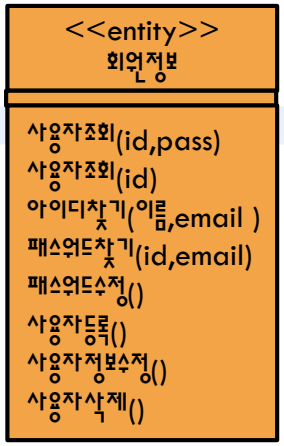


회원관리 클래스 다이어그램



LoginMapper.java interface 설계

```java
@Mapper //@Repository(value = "loginMapper")
public interface LoginMapper {

    //UserVO findUser(@Param(value = "userId") String userId, @Param(value = "password") String password);
    UserVO findUser(UserVO userVO);

    UserVO getUser(String userId); //ID 중복확인

    int updateUser(UserVO userVO); //정보수정
    int insertUser(UserVO userVO); //회원가입

    /*
    User findUser(String userId);
    User findId(String name);
    User findId(String userId); //ID 중복확인
    User findPass(UserVO userVO);

    void updateUser(User user);
    void insertUser(User user);
    void deleteUser(UserVO userVO);
   */
}
```



LoginMapper.xml 작성

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

<mapper namespace="com.jang.bbs.mapper.LoginMapper">
   <select id="findUser" parameterType="UserVO" resultType="UserVO">
        SELECT  userid, passwd, name
        FROM    bbs_member
        WHERE   userid = #{userId} AND passwd = #{passwd}
   </select>
   <select id="getUser" parameterType="String" resultType="UserVO">
       SELECT   *
        FROM    bbs_member
        WHERE   userid = #{userId}
    </select>

    <insert id="insertUser" keyProperty="mno">
        <selectKey keyProperty="mno" resultType="int" order="BEFORE">
            SELECT COALESCE(MAX(mno), 0) + 1 AS mno
            FROM   bbs_member
        </selectKey>

        INSERT INTO bbs_member (mno, name, passwd, userid, jumin, email )
        VALUES ( #{mno}, #{name}, #{passwd}, #{userId}, #{jumin}, #{email})
    </insert>

    <update id="updateUser" parameterType="UserVO">
        UPDATE  bbs_member
         SET    passwd = #{passwd}, email = #{email}
        WHERE   mno  = #{mno}
    </update>
```

LoginService interface 와 구현체 작성

```java
package com.jang.bbs.service;

import com.jang.bbs.model.UserVO;

public interface LoginService {

    UserVO findUser(UserVO userVO);

    UserVO getUser(String userId);

    int insertUser(UserVO userVO);

    int updateUser(UserVO userVO);

}
```

사용자조회(id,pass)
사용자조회(id)
아이디찾기(이름,email)
패스워드찾기(id,email)
패스워드수정()
사용자등록()
사용자정보수정()
사용자삭제()

```java
package com.jang.bbs.service;

import com.jang.bbs.model.UserVO;

public class LoginSeviceImpl implements LoginService {

    @Override
    public UserVO findUser(UserVO userVO) {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public UserVO getUser(String userId) {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public int insertUser(UserVO userVO) {
        // TODO Auto-generated method stub
        return 0;
    }

    @Override
    public int updateUser(UserVO userVO) {
        // TODO Auto-generated method stub
        return 0;
    }

}
```

LoginServiceImpl 기본 작성

```java
@Service(value = "loginService")
public class LoginServiceImpl implements LoginService {

    @Resource(name = "loginMapper") //autowired
    private LoginMapper loginMapper;

    @Override
    public UserVO findUser(UserVO userVO) {
        return loginMapper.findUser(userVO);
    }

    @Override
    public UserVO getUser(String userId) {
        return loginMapper.getUser(userId);
    }

    @Override
    public int insertUser(UserVO userVO) {
        return loginMapper.insertUser(userVO);
    }

    @Override
    public int updateUser(UserVO userVO) {
        return loginMapper.updateUser(userVO);
    }

}
```

LoginController

```java
@Controller
@RequestMapping("/member")
public class LoginController {

	@Autowired // @Resource(name = "loginService") bean 주입
	private LoginService loginService; //의존관계 설정

	@RequestMapping(value = "/login.do", method = RequestMethod.GET)
	public String login() {
		return "/member/login";
	}

	@RequestMapping(value="/login.do", method = RequestMethod.POST)
	public String loginProc(@Valid UserVO userVO, BindingResult result, Model model, HttpSession session)
	{

		if (result.hasFieldErrors("userId")||result.hasFieldErrors("passwd"))  {
			model.addAllAttributes(result.getModel());
			return "/member/login";
		}

		UserVO loginUser = this.loginService.findUser(userVO);

		if(loginUser == null){
			model.addAttribute("userId", "");
			model.addAttribute("errCode", 1); //not exist userId in database
			return "/member/login";
		} else {
			model.addAttribute("loginUser", loginUser);
			session.setAttribute("userId", loginUser.getUserId()); //세션에 변수등록
			session.setAttribute("userName", loginUser.getName()); //세션에 변수등록
			return "redirect:/board/list.do";
		}
	}
}
```

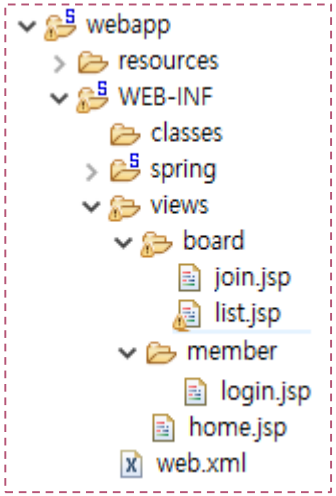
```java
public class UserVO {


	private int mno;

	@NotEmpty(message="id를 입력하세요!")
	private String userId;
	@NotEmpty(message="패스워드를 입력하세요!")
	private String passwd;
```

1. Login form에서 전달된 데이터를 UserVO DTO 객체에 넣고(@ModelAttribute 생략 가능) @Valid에 의해서 error 검사
2. 검사 결과를 BindingResult 객체에 저장

```java
@RequestMapping("/logout.do")
public String logout(HttpSession session){
	session.invalidate();
	return "redirect:/member/login.do";
}
```

Login.jsp

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"  pageEncoding="UTF-8"%>
<%@ taglib uri="http://www.springframework.org/tags" prefix="spring" %>
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Login</title>
<link href="${pageContext.request.contextPath}/resources/css/main.css" rel="stylesheet" type="text/css">
<c:if test="${errCode == null}">
          <c:set var="errCode" value="\"\""></c:set>
</c:if>
<script type="text/javascript">
     function checkErrCode(){
          var errCode = ${errCode};
          if(errCode != null || errCode != ""){
               switch (errCode) {
               case 1:
                    alert("가입된 이메일 주소나 비밀번호가 일치하지 않습니다!");
                    break;
               case 2:
                    alert("아이디나 비밀번호가 입력되지않았습니다.");
                    break;
               case 3:
                    alert("회원가입 처리가 완료되었습니다! 로그인 해 주세요!");
                    location.href = "<%=request.getContextPath()%>/login.do";
                    break;
               }
          }
     }
</script>
</head>
```



Login.jsp(2)

//controller 에서 보내온 errcode 메시지를 JSP페이지에서 출력

```
<body onload="checkErrCode()">
<div class="wrapper">
<h3>스프링 게시판</h3>

<form:form modelAttribute="userVO" method="post" action="lo
    <fieldset>
    <label for="userId">아 이 디 : </label>
    <input type="text" id="userId" name="userId" class="loginInput" value="${userId}" />
    <span class="error"><form:errors path="userId" /></span><br />
    <label for="userPw">비밀번호 : </label>
    <input type="password" id="passwd" name="passwd" class="loginInput"/>
    <span class="error"><form:errors path="passwd" /></span><br /><br />
    <center>
    <input type="submit" value="로그인" class="submitBt"/><br /><br />
    <a href="<%=request.getContextPath()%>/member/join.do">회원가입</a>
    <a href="<%=request.getContextPath()%>/member/findId.do">/아이디찾기</a>
    <a href="<%=request.getContextPath()%>/member/findPass.do">/비밀번호찾기</a>
    </center>
    </fieldset>
</form:form>
</div>

</body>
</html>
```

```
public class UserVO {

    private int mno;

    @NotEmpty(message="id를 입력하세요!")
    private String userId;
    @NotEmpty(message="패스워드를 입력하세요!")
    private String passwd;
```

//validatior에서 얻어진 메시지를 JSP페이지에서 출력

Java Script

- 스크립트(scripts)는 연극 용어 scripts(대본) 에서 유래 되었음
- Javascript - Netscape Communications 가 개발한 스크립트 언어이며. 웹 브라우저에서만 실행되는 스크립트 언어
  - HTML과 CSS로 만들어진 웹페이지를 동적으로 만들어 주는언어
- 2008 Google이 Chrome의 성능향상을 위해 javascript 엔진(Google V8 엔진) 개발
  - 웹 브라우저를 만드는 데 기반을 제공하는 오픈 소스 자바스크립트 엔진
- 2009 Ryan Dal 브라우저 안에서 뿐만이 아니라 서버에서도 작동되는 node.js 개발
  - Node.js - V8 자바스크립트 엔진으로 빌드 된 자바스크립트 런타임 라이브러리 . 이벤트기반, 논블로킹 I/O모델을 사용해 가볍고 효율적

- 웹 브라우저에서 동적인 사용자 인터페이스를 만들기 위해 사용되는 자바스크립트 라이브러리 (front-end 기술) 발전
  - Jquery, Angular, React, Vue 등

jQuery

- 빠르고, 작고, 기능이 풍부한 JavaScript 라이브러리
- HTML 문서 탐색 및 조작 , 이벤트 처리 , 애니메이션, Cross 브라우징
- 존 레식이 2006년 뉴욕(NYC) Barcamp 에서 공식적으로 소개

- jQuery의 기능
  - HTML 엘리먼트 선택 및 조작
  - HTML 이벤트 처리
  - CSS 셀렉터에 기반한 DOM 조작
  - 특수효과 및 애니메이션
  - AJAX
  - 멀티브라우저 지원 크로스브라우저
  - jQueryUI : jQuery를 기본으로 하면서 뷰를 표현하는 다양한 컴포넌트, 위젯을 모아놓은 User Interface를 모아 놓은 것

Jquery 문법

1. 기본문법

- $(A).action(B) : A 라는 HTML 문서의 특정 요소(태그)가 action이라는 행동을 했을 때 B 를 실행하라" : $ == jquery
  - $(this).hide() – 현재 요소를 화면에서 숨김
  - $("p").hide() – <p> 요소를 화면에서 모두 숨김
  - $(".test").hide() – class=" test" 라는 속성을 가진 요소를 찾아 화면에서 모두 숨김
  - $("#test").hide() – id=" test" 인 요소를 찾아 화면에서 숨김

2. 문서가 준비되었는지 확인한 후, 익명함수 function abc ()을 실행

익명(Anonymous) 함수 - 이름이 없고 일회용 으로 즉시 실행되는 함수

```
$(document).ready( function(){
        // jQuery 함수와 문법들이 이곳에 들어감
} );
```

ex)

```
$(document).ready( function() { alert("로딩 완료"); } );
```

Jquery 문법

3. 이벤트

```
$("p").click( function(){  $(this).hide(); } );
```

4. animate

```
$( A ).animate({ B }, speed, callback);
```

A라는 요소는 B 위치에 위치와 크기, 투명도 등의 css코드를 넣어주게 되면 해당 크기와 위치, 투명도로 애니메이션이 된다. speed를 넣어주면 속도가 결정된다.

```
$(  "button"  ).click(function(){
  $(  "div"  ).animate({left:'  250px'  });
});
```

Ajax 개요

- Asynchronous(비동기) Javascript And Xml
  - 브라우저의 XMLHttpRequest 객체를 이용해서 자바스크립트 내부에서 서버와 통신
  - 통신 후 리턴 받는 데이터는 xml, json, text 등 타입
  - 수신 데이터로 javascript에서 ui조작을 하여 전체 갱신없이 ui의 일부분만 동적 변경 가능
  - 페이지요청 후 화면이 전체 갱신되면서 html이 랜더링되는 일반적인 웹프로래밍 방법과는 다름
- 장점 :
  - Ajax 통신 후 받은 데이터를 이용 화면의 전체 갱신없이 부분적인 ui 변경 가능
  - html 전체코드가 아닌 최소한의 필요한 데이터(xml or json)만 받으므로 통신량 감소하고 Cross Browsing이 용이
- 단점 :
  - javascript 코딩 량 증가 .
- jquery를 이용한 ajax :
  - 일반 javascript ajax는 코딩량도 많아지고 브라우저 별로 구현방법이 다름
  - Jquery ajax 를 이용하면 더 적은 코딩량과 동일한 코딩방법으로 여러 브라우저에서 같은 동작을 할 수 있게 되는 Cross Browsing이 용이

Ajax 기본 문법

```
<script src="//code.jquery.com/jquery.min.js"></script>

<script>
$(document).ready( function() {
    $.ajax({
        url: "AddServer.jsp",
        type: "POST",
        data: $("#ajaxform").serialize(),
        dataType:  'json'  ,
        success : function(data) {
            // 통신이 성공적으로 이루어졌을 때
            수신된 데이터를 이용한 HTML 조작
        },
        error: function(e){
            alert('Error: ' + e);
        }
    });
});
</script>
```

JavaScript runtime library

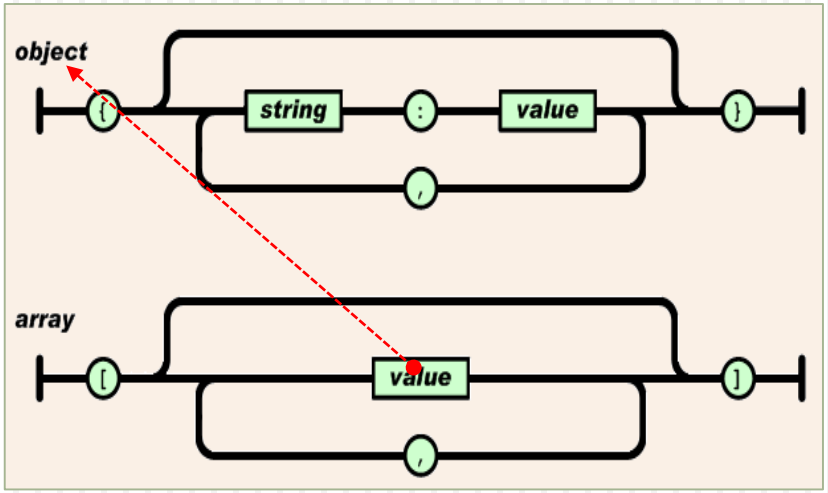
- url : 통신을 원하고자 하는 URL주소를 입력.(필수 입력 값)
- data : 서버로 보낼 데이터를 입력.
- type : GET, POST등의 전송 통신 방식 지정.
- dataType : 통신의 결과로 넘어올 수신 데이터의 종류를 지정.
- success(data) : 통신 성공시 호출 해야하는 함수를 지정. 매개변수로 결과로 넘어온 데이터를 수신.
- error : 통신 실패시 호출 해야하는 함수를 지정.
- complete : 통신 성공 여부와 관계없이 통신이 끝난 후 호출 해야하는 함수를 지정.
- beforeSend : 통신 전에 호출 해야하는 함수를 지정.
- async : 비동기(true), 동기(false) 여부를 지정.

ajax 예

```
//dictionary.js 구현
$(document).ready(function() {
    $('#ajax').click(function() {
        $.ajax({
            url:'ajax.xml',
            type:'GET',
            dataType: 'xml',
            success: function(data){
                $('#dictionary').empty();
                $.each($(data).find('entry'), function(){
                    var $entry = $(this);
                    var html ='<div class="entry">';
                    html +='<h3 class="term">'+ $entry.attr('term'); +'</h3>';
                    html +='<div class="part">'+ $entry.attr('part'); +'</div>';
                    html +='<div class="definition">'+  $entry.text()+'</div>';
                    html +='</div>';
                    $('#dictionary').append(html);
                });// end each
            }// end
        });// end ajax
        return false;
    });
});
```

Json 개요

- JSON (JavaScript Object Notation)은 경량의 DATA-교환 형식
- 속성:값 쌍 형식으로 사람과 기계가 분석하고 생성하기 용이.
- JSON 포맷은 더글라스 크록포드가 규정, RFC 7159와 ECMA-404 두 개의 표준존재
- JSON은 프로그래밍 언어로 부터 독립적
- JSON의 공식 인터넷 미디어 타입은 application/json이며, JSON의 파일 확장자는 .json이다.
- JSON의 기본 구조:



Map 등으로 구현

{"msg":"SUCCESS","education":"1234","name":"jjin"}

Allay list 로 실현

[{"msg":"SUCCESS","education":"1234","name":"jjin"},
{"msg": "FAIL","education": "1111","name": "kim"}
]

참고 싸이트 : https://www.json.org/json-ko.html

Ajax 연습

사용자를 추가해주세요(Ajax).

사용자명 : jjin

이 메 일 : 1234

사용자추가

Post

서 버

Name – jjin

E-mail - 1234

{"msg":"SUCCESS", "email":"1234","name":"jjin"}

사용자를 추가해주세요(Ajax).

사용자명 :

이 메 일 :

사용자추가

1. **Name** : jjin :: **Education** : 1234
2. **Name** : kim :: **Education** : 3456

json

json	{"msg":"SUCCESS", "email":"1234","name":"jjin"}
XML	<login> <msg>SUCCES</msg> <email>1234</email> <name>jjin</name> </login>

Hash Map {email=1234, name=jjin, msg=SUCCESS}

Ajax 실습

- File >new> Dynamic Web Project 생성 (ajaxTest)
- WebContent>WEB-INF>lib에 JSON 객체 생성을 위한 라이브러리 복사 (json_simple-1.1.jar or gson-2.8.5.jar)
- WebContent에 ajaxUser.jsp 파일 생성
- WebContent에 ajaxServer.jsp 파일 작성

- ajaxUser.jsp Run As
- 결과 확인

AjaxUser.jsp - Ajax Script

```
<%@ page language="java" contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Add Users Server for ajax</title>

      ~~~~~ Ajax script 삽입 ~~~~~

</head>
<body>
<h3>사용자를 추가해주세요(Ajax).</h3>
<form class="ajaxform" id="ajaxform" name="ajaxform" method="post" >
<table>
  <tr><td colspan="2"><div id="error" class="error"></div></td></tr>
  <tr><td>사용자명 : </td><td> <input type="text" id="name" name="name"><br/></td></tr>
  <tr><td>이  메  일 : </td><td> <input type="text" id="email" name="email"><br/></td></tr>
  <tr><td colspan="2"><input type="button" value="사용자추가" onclick="doAjaxPost()"><br/></td></tr>
  <tr><td colspan="2"> <div id="info" class="success"><ol></ol></div> </td></tr>
</table>
</form>

</body>
</html>
```

다음 페이지
ajax함수 추가

사용자를 추가해주세요(Ajax).

사용자명 :

이 메 일 :

사용자추가

AjaxUser.jsp - Ajax Script

```
<script src="//code.jquery.com/jquery.min.js"></script>
<script>
function doAjaxPost() {

    $.ajax({
      type: "POST",
      url: "ajaxServer.jsp",
      data: $("#ajaxform").serialize(),
      dataType:'json',
      success: function(response){
          alert('success==>' + response );
          if(response.msg == "SUCCESS"){

              $("#info ol").append("<br><li><b>Name</b> : " + response.name + " ➔ <b> E-mail</b> : "
                + response.email);
              $('#name').val('');
              $('#email').val('');
              $('#error').hide('slow');
              $('#info').show('slow');

          }else{
              errorInfo = "";
              for(i =0 ; i < response.result.length ; i++){
                  errorInfo += "<br>" + (i + 1) +". " + response.result[i].code;
              }
              $('#error').html("Please correct following errors: " + errorInfo);
              $('#info').hide('slow');
              $('#error').show('slow');
          }
        },
        error: function(e){
          alert('Error: ' + e);
        }
      });
  }
</script>
```

사용자를 추가해주세요(Ajax).

사용자명 :

이 메 일 :

사용자추가

response: {"name":"jjin","email":"1234","msg":"SUCCESS"}

```
<div id="info" class="success"><ol></ol></div>
```

AjaxServer.jsp -json, gson

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page import="java.util.*"%>
<%@ page import="org.json.simple.*"%>
<%@ page import="com.google.gson.*"%>

<%
    request.setCharacterEncoding("utf-8");
    response.setContentType("application/json; charset=utf-8");

    String name = request.getParameter("name");
    String email = request.getParameter("email");

    Gson gson = new Gson();

    JsonObject object = new JsonObject();

    object.addProperty("name", name);
    object.addProperty("email", email);
    object.addProperty("msg","SUCCESS");

    String json = gson.toJson(object);

    response.getWriter().write(json.toString());

    //System.out.println(json);
%>
```

gson : JSON 객체 ←→ java 객체를 돕는 라이브러리

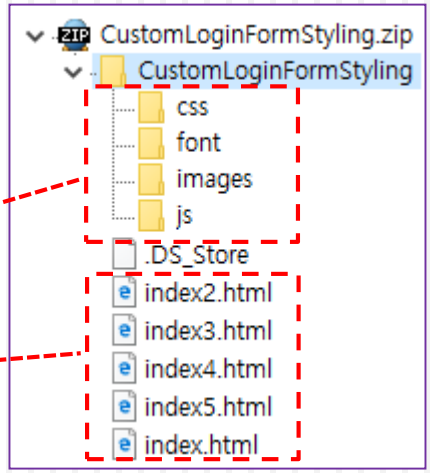
- ajaxtest
  - Deployment Descriptor: ajaxtest
  - JAX-WS Web Services
  - Java Resources
  - JavaScript Resources
  - build
  - WebContent
    - css
    - font
    - images
    - META-INF
    - WEB-INF
      - lib
        - gson-2.8.5.jar
        - json-simple-1.1.1.jar
      - ajaxlogin.jsp
      - AjaxServer.jsp
      - AjaxUser.jsp

```java
JSONArray respJson = new JSONArray();
JSONObject  obj = new JSONObject();

obj.put("name", name);
obj.put("email", email);
obj.put("msg","SUCCESS");

response.getWriter().write(obj.toString()); http 페이지에 출력
//{"msg":"SUCCESS","education":"1234","name":"jjin"}

respJson.add(obj); // {} 을 list 형태로 만들때
respJson.toString(); //[{"msg":"SUCCESS","education":"1234","name":"jjin"}]
```

```
{"name":"jjin","email":"1234","msg":"SUCCESS"}
```

서버사이드 spring 인 경우 -게시판 Login 예제

- 로그인 폼 5개 다운로드
- http://tympanus.net/codrops/2012/10/16/custom-login-form-styling/
- Dynamic Web Project 인 경우
  - Webcontent에 복사
  - HTML에서 배경 제거
- 스프링 Project 경우 경로변경
  - Webapp/resource에 복사
  - view/member/에 복사
  - New>jsp ➔ ajaxlogin.jsp 생성
  - Index2.html 파일 내용 전체 복사 ➔ ajaxlogin.jsp 파일에 추가
  - Pom 추가




```
<dependency>
    <groupId>com.google.code.gson</groupId>
    <artifactId>gson</artifactId>
    <version>2.8.5</version>
</dependency>
```

ajaxlogin.jsp 수정 -게시판 Login 예제

1. *Ressource 경로 수정 ="${pageContext.request.contextPath}/resources/*

```
<link rel="stylesheet" type="text/css" href="${pageContext.request.contextPath}/resources/css/style.css" />
<script src="${pageContext.request.contextPath}/resources/js/modernizr.custom.63321.js"></script>
```

2. HTML과 demo.css 에서 (background:) 제거

```
<style>
    body {
        background: #e1c192 url(images/wood_pattern.jpg);
    }
</style>
```

3. ~ * *, <header>~<header> 제거

4. 폼변경

```
<body>
   <div class="container">
        <section class="main">
            <form:form modelAttribute="user" class="form-2" name="form-2" id="form-2" method="post" >
                <h1><span class="log-in">Log in</span> or <span class="sign-up">sign up</span></h1>
                <p class="float">
                    <label for="login"><i class="icon-user"></i>아이디</label>
                    <input type="text" name="userId" id="userId" placeholder="Username or email">
                </p>
                <p class="float">
                    <label for="password"><i class="icon-lock"></i>패스워드</label>
                    <input type="password" name="passwd" id="passwd" placeholder="Password" class="showpassword">
                </p>
                <p class="clearfix">
                    <a href="#" class="log-twitter">회원가입</a>
                    <input type="button" name="submit" value="로그인" onclick="formSubmit()" />
                </p>
            </form:form>
        </section>
    </div>
```

Ajaxlogin.jsp 처리

5. <head> ~ </head>에 Ajax 스크립트 삽입

```
<script>
function formSubmit() {

    $.ajax({
        url: 'ajaxlogin.do',
        type: 'POST',
        data: $("#form-2").serialize(),
        dataType: 'json',
        success: function (result) {

            $("#ajax").remove();
            if (result.msg=="Success"){
                  alert('환영합니다! ' + result.id+' 님');
                  location.href="http://localhost:8080/bbs/board/list.do";
            }
            else {
                   if (result.err=="eid"){
                       alert('ID가 입력되지않았습니다.');
                   }
                   else if(result.err=="epass"){
                       alert('패스워드가 입력되지않았습니다.');
                   }
                   else{
                       alert('ID 또는 패스워드가 잘못 입력되었습니다');
                   }

                  location.href ="http://localhost:8080/bbs/member/ajaxlogin.do";
            }
        }

    });
}
</script>
```

LoginController 에 /ajaxlogin.do 추가

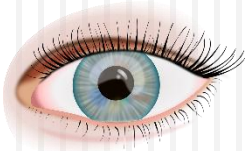
```
@RequestMapping(value= "/ajaxlogin.do", method=RequestMethod.POST)
public @ResponseBody String AjaxView(@Valid UserVO userVO, BindingResult bindingResult, HttpSession
session) {
    Gson gson = new Gson();
    JsonObject object = new JsonObject();
    object.addProperty("msg", "Fail");

    if (bindingResult.hasFieldErrors("userId"))  {
        object.addProperty("err", "empty_id");
        return gson.toJson(object).toString();
    }
    else if(bindingResult.hasFieldErrors("passwd")){
        object.addProperty("err", "empty_pass");
        return gson.toJson(object).toString();
    }

    UserVO loginUser = this.loginService.findUser(userVO);

    if(loginUser == null){
        object.addProperty("id", "Null");
        object.addProperty("err", "noUser");
        return gson.toJson(object).toString();
    }
    else {
        session.setAttribute("userId", loginUser.getUserId()); //세션에 변수등록
        session.setAttribute("userName", loginUser.getName()); //세션에 변수등록
                                    //Json Map==>{msg=Success, id=jjin}
        object.addProperty("id", loginUser.getUserId());
        object.addProperty("msg", "Success");
        object.addProperty("err", "");

        return gson.toJson(object).toString();
    }
}
```

```
@RequestMapping("/ajaxlogin.do")
public String ajlogin() {
        return "/member/ajaxlogin";
}
```

```
<dependency>
    <groupId>com.google.code.gson</groupId>
    <artifactId>gson</artifactId>
    <version>2.8.5</version>
</dependency>
```

관점(관심) 지향 프로그래밍

- AOP (Aspect-Oriented Programming)
  - 해당 소스에 있는 코드가 핵심기능을 담당하는지, 부가적인 공통적인 기능을 담당하는 코드인지 관심사를 구분하고
  - 문제를 해결하기 위한 핵심 관심 사항과 전체에 적용되는 공통 관심 사항을 기준으로 관심사들을 별도의 독립적인 엔티티로 그룹화하여 프로그래밍.
  - 공통기능 코드가 변경되면 다른 로직부분의 공통기능부분도 모두 수정을 해줘야 하지만 공통기능을 따로 분리하여 처리함으로써 핵심기능(핵심로직) 코드는 수정없이 변경된 공통기능의 코드만 적용하면 됨.



핵심 기능관점 : 공통점 없음

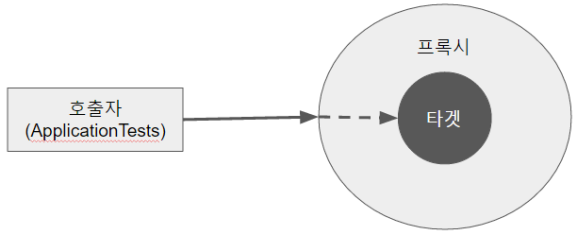
BoardService	UserService	XXXService
before()	before()	before()
getBoards()	getUsers()	getXXXs()
after()	after()	after()

공통부가 기능관점 : 전후에 처리해야 할 기능
공통점 (로그(사용기록), 보안(권한)

AOP (Aspect-Oriented Programming)

- AOP의 장점
  - 어플리케이션 전체에 흩어진 공통 기능이 하나의 장소에서 구현
  - 다른 서비스 모듈들이 본인의 목적에만 충실하고 그외 사항들은 고려하지 않아도 가능

- 2. AOP 용어
  - advice : 공통 기능을 구현한 것(클래스)
  - target : advice가 적용될 객체
  - JoinPoint : advice가 적용될 지점(target의 메소드)
  - pointcut : 실제로 advice가 적용된 joinpoint (pointcut은 joinpoint의 부분집합)
    advisor(= aspect) : advice + pointcut
    weaving : advice를 핵심 로직 코드에 적용하는 것
  - 프록시 (Proxy) :타겟을 감싸서 타겟의 요청을 대신 받아주는 랩핑(Wrapping) 오브젝트



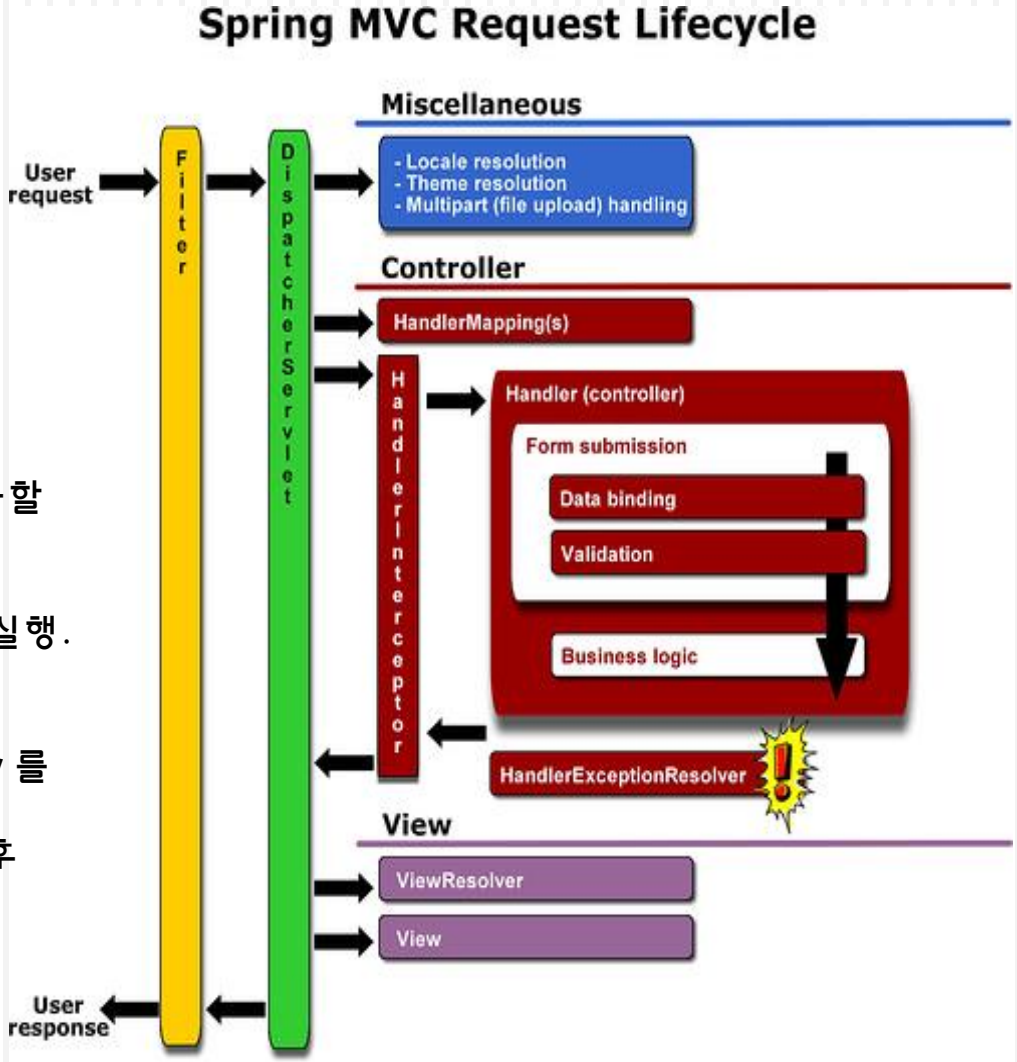
Interceptor, filter, AOP

- 스프링에서 사용되는 Filter, Interceptor, AOP 세 가지 기능은 모두 무슨 행동을 하기 전에 먼저 실행하거나, 실행한 후에 추가적인 행동을 할 때 사용되는 기능
- AOP의 경우 Interceptor나 Filter와는 달리 메소드 전후의 지점에 자유롭게 설정이 가능
- Interceptor와 Filter는 주소로 대상을 구분해서 걸러내야하는 반면, AOP는 주소, 파라미터, 애노테이션 등 다양한 방법으로 대상을 지정할 수 있다는 장점

구분	Filter	Interceptor	AOP
실행 위치	서블릿	서블릿	메서드
실행 순서	1	2	3
설정 위치	web.xml	xml or java	xml or java
실행 메서드	init(), doFilter(), destroy()	preHandler(), postHanler(), afterCompletion()	포인트 컷으로 @After, @Before, @Around 위치를 지정

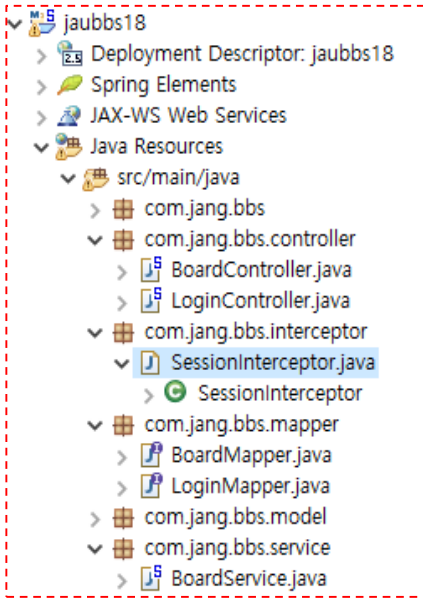
Interceptor

- intecept: (중간에) 가로막다, 가로채다
- 시나리오 : 어떠한 페이지를 접속하려고 할 때, 로그인된 사용자만 접속허용하고, 로그인이 되어있지 않다면 로그인 화면으로 이동시킨다.
- 문제점 : 페이지마다 로그인 체크 로직을 만들어서 처리하면 너무 복잡하다.
- 인터셉터는 DispatcherServlet이 컨트롤러를 호출하기 전,후에 요청과 응답을 가로채서 가공할 수 있도록 해준다.
  - preHandle : Controller 가 수행되기 전에 실행. 이후 Controller를 수행할지 여부를 boolean으로 return.
  - postHandle : Controller 가 수행된 후 View 를 호출하기 전 상태
  - afterCompletion : View 작업까지 완료된 후 호출



Interceptor 생성

```
public class SessionInterceptor extends HandlerInterceptorAdapter {
   @Override
   public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler)
                           throws Exception {

      Object userId = request.getSession().getAttribute("userId");

      // login.do 이나 회원가입페이지를 실행하려는 경우(/bbs/member/join.do) 이미 로그인 되어 있는 자는 로그인이나 가입이 필요없음==>list.do
      if(request.getRequestURI().equals("/bbs/member/login.do")||
            request.getRequestURI().equals("/bbs/member/join.do")||
            request.getRequestURI().equals("/bbs/member/ajaxlogin.do")) {
            if(userId != null){
               response.sendRedirect(request.getContextPath() + "/board/list.do");
               return true; //list.do 실행
            }
            else {
               return true; //login.do 실행
            }
       }
      if(userId == null){      //url이  /login or join.do 가 아닌 모든 경우 로그인 되어 있지않으면
          response.sendRedirect(request.getContextPath() + "/member/ajaxlogin.do");
          return false; //현재 URL 페이지 실행 안고 login.do 실행
      }
      else {
          return true; //현재 URL 페이지 실행
      }
   }
   @Override
   public void postHandle(HttpServletRequest request,HttpServletResponse response, Object handler,
                  ModelAndView modelAndView) throws Exception {
   }
}
```

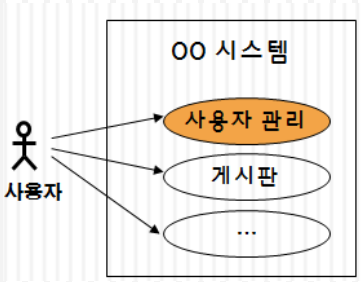


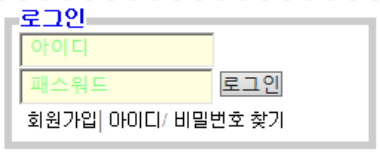
Servlet-Context.xml 에 interceptors 등록

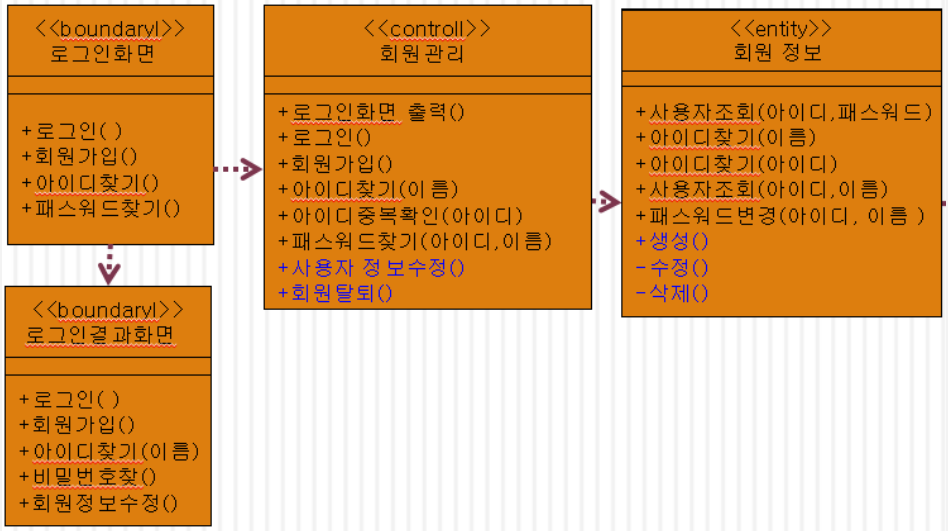
List 와 글보기(view.do)는 누구나 볼수 있어야 하므로 interceptor 제외

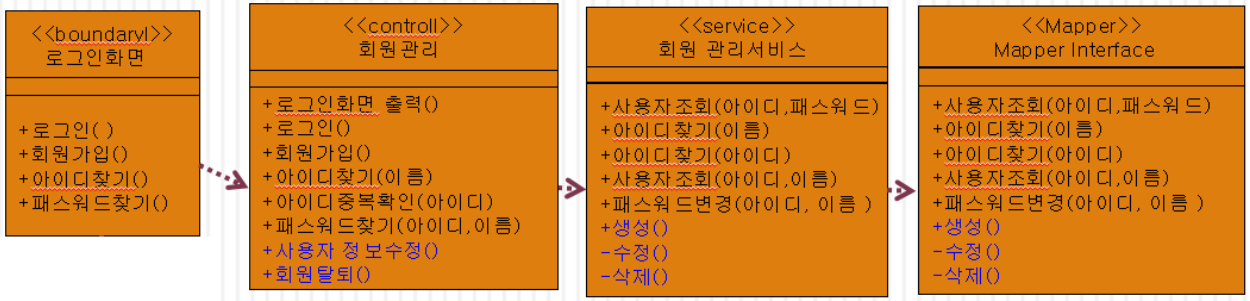
```
<interceptors>
    <interceptor>
        <mapping path="/**/*" />
        <exclude-mapping path="/board/list*" />
        <exclude-mapping path="/board/view*" />
        <beans:bean class="com.jang.bbs.interceptor.SessionInterceptor"></beans:bean>
    </interceptor>
</interceptors>
```

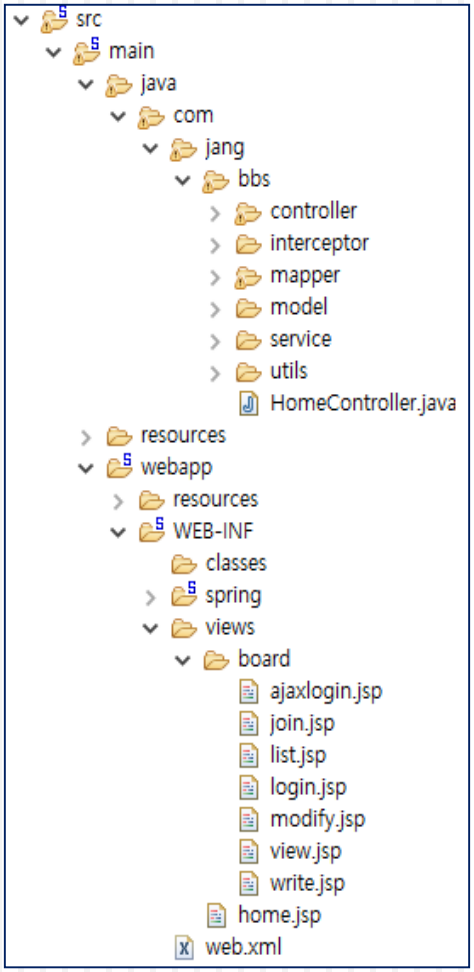
회원 관리 설계와 구현


OO 시스템
사용자 관리
게시판
...
사용자


로그인
아이디
패스워드
로그인
회원가입| 아이디/ 비밀번호 찾기


<<boundaryl>>
로그인화면
+로그인()
+회원가입()
+아이디찾기()
+패스워드찾기()
<<controll>>
회원관리
+로그인화면 출력()
+로그인()
+회원가입()
+아이디찾기(이름)
+아이디중복확인(아이디)
+패스워드찾기(아이디,이름)
+사용자 정보수정()
+회원탈퇴()
<<entity>>
회원 정보
+사용자조회(아이디,패스워드)
+아이디찾기(이름)
+아이디찾기(아이디)
+사용자조회(아이디,이름)
+패스워드변경(아이디, 이름)
+생성()
-수정()
-삭제()
<<boundaryl>>
로그인결과화면
+로그인()
+회원가입()
+아이디찾기(이름)
+비밀번호찾()
+회원정보수정()


<<boundaryl>>
로그인화면
+로그인()
+회원가입()
+아이디찾기()
+패스워드찾기()
<<controll>>
회원관리
+로그인화면 출력()
+로그인()
+회원가입()
+아이디찾기(이름)
+아이디중복확인(아이디)
+패스워드찾기(아이디,이름)
+사용자 정보수정()
+회원탈퇴()
<<service>>
회원 관리서비스
+사용자조회(아이디,패스워드)
+아이디찾기(이름)
+아이디찾기(아이디)
+사용자조회(아이디,이름)
+패스워드변경(아이디, 이름)
+생성()
-수정()
-삭제()
<<Mapper>>
Mapper Interface
+사용자조회(아이디,패스워드)
+아이디찾기(이름)
+아이디찾기(아이디)
+사용자조회(아이디,이름)
+패스워드변경(아이디, 이름)
+생성()
-수정()
-삭제()


src
main
java
com
jang
bbs
controller
interceptor
mapper
model
service
utils
HomeController.java
resources
webapp
resources
WEB-INF
classes
spring
views
board
ajaxlogin.jsp
join.jsp
list.jsp
login.jsp
modify.jsp
view.jsp
write.jsp
home.jsp
web.xml

실 행

글쓰기 : 파일 Upload

1. Pom에 fileupload <dependency> 추가
2. Servlet에 *multipartResolver 추가*
3. write.jsp View 작성
4. Controller – Write.do 작성

1. Pom 추가

```
<!-- Apache Commons FileUpload -->
<dependency>
    <groupId>commons-fileupload</groupId>
    <artifactId>commons-fileupload</artifactId>
    <version>1.3.3</version>
</dependency>

<!-- Apache Commons IO -->
<dependency>
    <groupId>commons-io</groupId>
    <artifactId>commons-io</artifactId>
    <version>2.6</version>
</dependency>
```

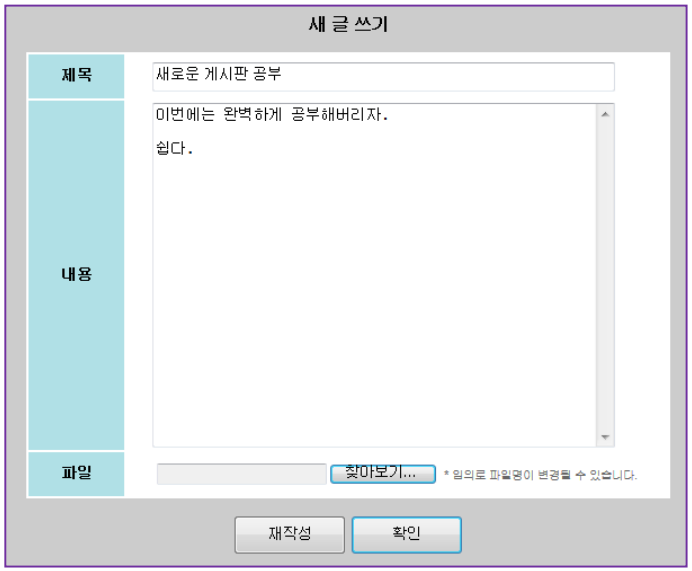
글쓰기 : multipartResolver추가(servlet –context.xml)

2.

```
<beans:bean id="multipartResolver"
      class="org.springframework.web.multipart.commons.CommonsMultipartResolver">
    <beans:property name="maxUploadSize" value="100000000" />
</beans:bean>
```

3. Write.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"  pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>새 글 쓰기</title>
<link href="${pageContext.request.contextPath}/resources/css/board.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" src="${pageContext.request.contextPath}/resources/js/jquery-1.7.1.js"></script>
<script type="text/javascript">
  function writeFormCheck() {
        if($("#title").val() == null || $("#title").val() == ""){
                alert("제목을 입력해 주세요!");
                $("#title").focus();
                return false;
        }
        if($("#content").val() == null || $("#content").val() == ""){
                alert("내용을 입력해 주세요!");
                $("#content").focus();
                return false;
        }
        return true;
  }
</script>
</head>
```



글쓰기 : multipartResolver추가(write.jsp)

3. Write.jsp

```
<body>
<div class="wrapper">
<h3>새 글 쓰기</h3>
<form action="write.do" method="post" onsubmit="return writeFormCheck()" enctype="multipart/form-data">
        <table class="boardWrite">
        <tr>
                <th><label for="subject">제목</label></th>
                <td>
                <input type="text" id="title" name="title" class="boardSubject"/>
                <input type="hidden" id="writer" name="writer" value="${userName}" /> <!-- 세션변수 -->
                <input type="hidden" id="writerid" name="writerid" value="${userId}" />
                </td>
        </tr>
        <tr>
                <th><label for="content">내용</label></th>
                <td><textarea id="content" name="content" class="boardContent"></textarea></td>
        </tr>
        <tr>
                <th><label for="file">파일</label></th>
                <td><input type="file" id="file" name="file" multiple />
                <span class="date">  * 임의로 파일명이 변경될 수 있습니다.</span></td>
        </tr>
        </table>
        <br />
        <input type="reset" value="재작성" class="writeBt"/>
        <input type="submit" value="확인" class="writeBt"/>
</form>
</div>
</body>
</html
```

writeFormCheck() 함수를 실행한 후 true 이면 submit

Login 에서 설정된 Session 변수 내용을 읽어옴

글쓰기 : BoardController에 (write.do) 추가

4.

```
@RequestMapping(value="/write.do", method=RequestMethod.GET)
public String boardWrite(){
return "/board/write";
}

@RequestMapping(value="/write.do", method=RequestMethod.POST)
public String boardWriteProc(@ModelAttribute("Board") BoardVO board,
        MultipartHttpServletRequest request){

    //새 글저장
    String content =  board.getContent().replaceAll("\r\n", "<br />");//java새줄 코드 HTML줄바꾸기로
    content = content.replaceAll("<","&lt;");
    content = content.replaceAll(">","&gt;");
    content = content.replaceAll("&","&amp;");
    content = content.replaceAll("\"","&quot;");

    board.setContent(content);

    boardService.writeArticle(board);

    int bno = board.getBno(); //저장시 생성한 새글번호  xml 파일insert 에서 keyProperty="bno" 에 의해서 설정됨
```

글쓰기 : BoardController에 (write.do) 추가

```
    //첨부 파일 저장
    AttFileVO file = new AttFileVO();
    String uploadPath = "C:\\upload\\";
         List<MultipartFile> fileList = request.getFiles("file");

         for (MultipartFile mf : fileList) {
              if (!mf.isEmpty()) {
                   String originFileName = mf.getOriginalFilename(); // 원본 파일 명
                   long fileSize = mf.getSize(); // 파일 사이즈

                   System.out.println("originFileName : " + originFileName);
                   System.out.println("fileSize : " + fileSize);

                   file.setBno(bno);
                   file.setOfilename(originFileName);
                   file.setSfilename(originFileName);
                   file.setFilesize(fileSize);

                   boardService.insertFile(file);//테이블에 화일 정보 저장

                  String safeFile = uploadPath +  originFileName; // 디스크에 파일 저장

                   try {
                       mf.transferTo(new File(safeFile)); // 디스크에 파일 저장
                   } catch (IllegalStateException e) {
                       e.printStackTrace();
                   } catch (IOException e) {
                       e.printStackTrace();
                   }
              }
         }
         return "redirect:list.do";
}
```

글보기

글보기 연습

작성자	조회수	추천수	작성일
TestId	1	0	2018-10-30 11:43

내용

첨부파일	golf.bak (93152 byte) oracle.txt (126 byte)

반드시 login 되어 있어야하고
댓글, 첨부파일 처리해야함

추천 0

댓글

TestId 삭제	2018-10-30 11:44 대글 입력 및 삭제, 추천 처리 추천 0
댓글 쓰기	확인

삭제 수정 목록

글보기: BoardController에 추가 (view.do)

```
@RequestMapping("/view")
public String boardView(@RequestParam(value = "bno", required = false, defaultValue="0")
int bno, HttpSession session, Model model) throws Exception {

    String userId = (String) session.getAttribute("userId");

    BoardViewVO boardViewVO = new BoardViewVO() ; //글조회 이력 용
    boardViewVO.setBno(bno);

    if(userId != null ) { //회원인 경우
         boardViewVO.setUserId(userId);
         boardViewVO.setMem_yn('y'); //회원

         if( boardService.getBoardView(boardViewVO) == 0) {//해당 번호 글을 읽은 기록이 없으면
              boardService.increaseViewCnt(bno); // 조회수 갱신
              boardService.addBoardView(boardViewVO); //회원 조회 등록
         }
    }
    else { //비회원인 경우
     boardViewVO.setUserId(session.getId()); //세션id를 회원 id로 등록
     boardViewVO.setMem_yn('n'); //비회원

     if(boardService.getBoardView(boardViewVO) == 0) {{//해당 번호 글을 읽은 기록이 없으면
          boardService.increaseViewCnt(bno); // 조회수 갱신
          boardService.addBoardView(boardViewVO);//비회원 조회수 등록
     }
   }
```

```
public class BoardViewVO {
    private int bvno;
    private int bno;
    private String userId;
    private char mem_yn;
    private String regDate;
}
```

글보기: BoardController에 추가 (view.do)

```
    BoardVO board = boardService.getArticle(bno); // get selected article model

    List<ReplyVO> reply = boardService.getReplyList(bno); // 답변 목록 읽어 오기 – list
    List<AttFileVO> fileList = boardService.getFileList(bno); // 첨부파일 목록 읽어 오기-list

    model.addAttribute("board", board);
    model.addAttribute("replyList", reply);
    model.addAttribute("fileList", fileList);

    return "board/view";
}
```

Service

```
boardService.getBoardView(boardViewVO)
boardService.increaseViewCnt(bno);
boardService.addBoardView(boardViewVO);
boardService.getArticle(bno)
boardService.getReplyList(bno)
boardService.getFileList(bno)
```

```
public int addBoardView(BoardViewVO boardView){   //userid-글번호 조회 기록 추가
    return this.boardMapper.addBoardView(boardView);
}
public  int increaseViewCnt(int bno){  //글번호 조회수 증가
    return this.boardMapper.incrementViewCnt(bno);
}
public int getBoardView(BoardViewVO boardView){ //userid-글번호 조회 기록 읽기
    return this.boardMapper.getBoardView(boardView);
}

public BoardVO getArticle(int bno){ //글번호 글읽기
    return this.boardMapper.getArticle(bno);
}
public List<ReplyVO> getReplyList(int bno) {//글번호-답글 리스트 읽어 오기
    return this.boardMapper.getReplyList(bno);
}
public List<AttFileVO> getFileList( int bno) {//글번호-첨부 파일 리스트 읽어오기
    return this.boardMapper.getFileList(bno);
}
```

글보기 Mapper

```
<select id="getArticle" resultType="BoardVO">
   SELECT  bno , title, content, writerId, to_char(regDate, 'YYYY-MM-DD H
    FROM   bbs_board
    WHERE  bno = #{bno}
</select>
```

```
<insert id="addBoardView" keyProperty="bvno">
    <selectKey keyProperty="bvno" resultType="int" order="BEFORE">
        SELECT COALESCE(MAX(bvno), 0) + 1 AS bvno
        FROM   bbs_boardview
    </selectKey>

    INSERT INTO bbs_boardview(bvno, bno, userId, mem_yn )
    VALUES ( #{bvno}, #{bno}, #{userId}, #{mem_yn})
</insert>

<select id="getBoardView" resultType="int">
    SELECT COALESCE(MAX(bno), 0) AS bno
    FROM   bbs_boardview
    WHERE userId = #{userId} AND bno = #{bno}
</select>
```

```
@Repository(value = "boardMapper") //@Mapper
public interface BoardMapper {

    List<BoardVO> getBoardList(SearchVO searchVO); //글 목록 조회

    BoardVO getArticle(int bno);       //글조회
    int writeArticle(BoardVO board); //글저장
    int updateArticle(BoardVO board); //글수정
    void deleteArticle(int bno);      //글삭제

    int getTotalArticle(SearchVO searchVO); //전체 글수 조회

    int incrementViewCnt(int bno);  //조회수 증가
    int incrementGoodCnt(int bno);  //좋아요 증가
    int incrementReplyCnt(int bno); //댓글수 증가

    List<ReplyVO> getReplyList(int bno);  //댓글 목록 조회
    ReplyVO getReply(int rno);  // 댓글 조회

    int writeReply(ReplyVO reply); //댓글 저장
    int updateReply(ReplyVO reply); //댓글 수정
    void deleteReply(int rno);       //댓글 삭제
    void deleteReplyBybno(int bno); //원글에 소속된 댓글 전체 삭제

    int incReplyGoodCnt(int rno); //댓글 좋아요 증가
    // int incrementReMultiCnt(int bno);

    List<AttFileVO> getFileList( int bno); //첨부파일 목록 읽기
    String getFileName(int fno);  //파일 이름 조회
    int insertFile(AttFileVO file); //파일 저장

    void deleteFile(int fno);  //첨부 파일 삭제
    void deleteFileBybno(int bno); //원본 소속된 첨부파일 전체 삭제

    int addBoardLike(BoardLikeVO boardLike); //원본글 추천 기록 저장
    int getBoardLike(BoardLikeVO boardLike); //원본글 추천 기록 확인

    int addReplyLike(ReplyLikeVO replyLike); //댓글 추천기록 저장
    int getReplyLike(ReplyLikeVO replyLike); //댓글 추천 기록 확인

    int addBoardView(BoardViewVO boardView); //원본글 추천 기록 저장
    int getBoardView(BoardViewVO boardView); //원본글 추천 기록 확인
```

글보기 – view.jsp 작성

```jsp
        <td >첨부파일</td>
        <td colspan="4" align="left" >
            <c:forEach var="file" items="${fileList}">
                <a href="filedown.do?fileName=${file.ofilename}" class="fileview"><font size="2px">${file.ofilename}</font>
                <font size="2px">(${file.filesize} byte)</font><br> </a>
            </c:forEach>
            <c:if test="${empty fileList}">
                <font color="#A6A6A6" size="2px"> 첨부된 파일이 없습니다. </font>
            </c:if>
        </td>
    </tr>

    <tr>
        <td colspan="4" align="left"><p>${board.content}</p><br /><br />
        <c:if test="${board.writerId != userId}">
              <p align="center" > <a href="#" onclick="moveAction(4)"><small style="color:red" >추천 </small></a><span><small style="color:red" >:${message}</small></span> </p>
        </c:if>

        </td>
    </tr>
```

```jsp
<table class="commentView">
    <tr><th colspan="2">댓글</th></tr>
    <c:forEach var="comment" items="${replyList}">
    <tr>
        <td class="writer">
            <p>${comment.writerId}
            <c:if test="${comment.writerId == userId}">
                <br /><a href="#" onclick="replyDelete(${comment.rno}, ${comment.bno})"><small align="center" style="color:red">삭제</small></a>
            </c:if>
            </p>
        </td>
        <td class="content" align="left">
            <span class="date">${comment.regDate}</span>
            <p>${comment.content}</p>
            <c:if test="${comment.writerId != userId}">
              <p align="center" > <a href="#" onclick="recommandReply(${comment.rno}, ${comment.bno})"><small style="color:red" >추천 ${comment.recommendcnt}</small></a>
              </span><small style="color:red" >:${message}</small></span></p>
            </c:if>

        </td>

    </tr>
    </c:forEach>
```

글수정(1):BoardController에 추가 (modify.do 수정화면)

```
@RequestMapping(value = "/modify.do", method=RequestMethod.GET )
public String boardModify(HttpServletRequest request, HttpSession session, Model model){

    String userId = (String) session.getAttribute("userId");
    int bno = Integer.parseInt(request.getParameter("bno"));

    BoardVO board = boardService.getArticle(bno);

    // <br /> tag change to new line code
    String content = board.getContent().replaceAll("<br />", "\r\n");
    board.setContent(content);

    if(!userId.equals(board.getWriterId())){ //비회원 글수정 불가
        model.addAttribute("errCode", "1");
        model.addAttribute("bno", bno);
        return "redirect:view.do";
    } else {//회원 글수정
        List<AttFileVO> fileList = boardService.getFileList(bno); // 첨부파일 읽어 오기 - list

        model.addAttribute("board", board);
        model.addAttribute("fileList", fileList);
        return "/board/modify";
    }
}
```

글수정 modify.jsp

- BoardService ➔ BoardMapper.XML(::Mapper.Java)확인

```
    <tr>
        <th>내용</th>
        <td align="left"><textarea id="content" name="content" class="boardContent">${board.content}</textarea></td>
    </tr>
    <tr>
        <th><label for="file">첨부파일</label></th>
        <td align="left">
            <c:forEach var="file" items="${fileList}">
                <p>  <input type="checkbox" name="fileno" value="<c:out value="${file.fno}"/>">
                <c:out value="${file.ofilename}"/> <c:out value="(${file.filesize} byte)"/><span class="date">  *  ✓-삭제</span>
                    </p>
            </c:forEach>
            <c:if test="${empty fileList}">
                <font color="#A6A6A6" size="2px"> 첨부된 파일이 없습니다. </font>
            </c:if>
        </td>
    </tr>  ✓



    <tr>
        <th><label for="file">파일수정</label></th>
        <td align="left">><input type="file" id="file" name="file" multiple />><span class="date">  * 파일명이 변경될 수 있습니다.</span></td>
    </tr>

</table>
```

글수정(2): BoardController 추가 (modify.do 수정 저장후 view로)

```
@RequestMapping(value = "/modify.do", method=RequestMethod.POST)  //게시판 글 수정
public String  boardModifyProc(@ModelAttribute("Board") BoardVO board, MultipartHttpServletRequest
request, RedirectAttributes rea ){

      String content =  board.getContent().replaceAll("\r\n", "<br />"); //java 새줄 코드 HTML줄바꾸기로
      board.setContent(content);

      boardService.updateArticle(board);

      int bno = board.getBno();


      //체크된 파일을 테이블과 디스크에서 삭제한다.
      String[] fileno = request.getParameterValues("fileno");

      if (fileno != null) {

            for (String fn : fileno) {

                  int fno = Integer.parseInt(fn);

                  String oFileName = boardService.getFileName(fno);

                  String safeFile = uploadPath +  oFileName;

                  File removeFile = new File(safeFile);// remove disk uploaded file
                  removeFile.delete();

                  boardService.deleteFile(fno); //remove table uploaded file
            }
      }
```

글수정(2): BoardController 추가 (modify.do 수정 저장후 view로)

```
        AttFileVO file = new AttFileVO ();

        //새첨부 파일  목록 일어오기
        List<MultipartFile> fileList = request.getFiles("file");

        for (MultipartFile mf : fileList) {
            if (!mf.isEmpty()) {
                String originFileName = mf.getOriginalFilename(); // 새첨부파일 원본 파일 명
                    long fileSize = mf.getSize(); // 파일 사이즈

                file.setBno(bno);
                file.setFilesize(fileSize);
                file.setOfilename(originFileName);
                file.setSfilename(originFileName);

                boardService.insertFile(file); // 테이블에 파일 저장

                String safeFile = uploadPath +  originFileName;

                try {
                    mf.transferTo(new File(safeFile)); // 디스크에 파일 저장
                } catch (IllegalStateException e) {
                    e.printStackTrace();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
        rea.addAttribute("bno", board.getBno());
        return "redirect:/board/view.do";
    }
}
```

글삭제 : BoardController에 추가 (delete.do)

```
@RequestMapping("/delete.do")
public String boardDelete(HttpServletRequest request, HttpSession session, RedirectAttributes rea){
      String userId = (String) session.getAttribute("userId"); // login 개발뒤 삭제
      int bno = Integer.parseInt(request.getParameter("bno"));

      BoardVO board = boardService.getArticle(bno);

      String setView=""
      if(userId.equals(board.getWriterId())){
            //답글 삭제
            List<ReplyVO> reply = boardService.getReplyList(bno);
            if(reply.size() > 0){
                        boardService.deleteReplyBybno(bno);
            }
            //첨부 파일명 삭제, 실제 파일 삭제
            List<AttFileVO> files  = boardService.getFileList(bno);
            if(files.size() > 0){
                  //저장된  실제 파일 삭제
                  for (AttFileVO filedel : files) {
                        String f_stor_all = filedel.getOfilename();
                        File f = new File(session.getServletContext().getRealPath("/") + f_stor_all);
                        f.delete();
                  }
                  boardService.deleteFileBybno(bno); //테이블에서 해당 번호 글 첨부 file 전체 삭제
            }
            //board 삭제
            boardService.deleteArticle(bno)
            setView="redirect:list.do";
      } else {
            rea.addAttribute("errCode", "1");// it's forbidden connection
            rea.addAttribute("bno", bno);
            setView="redirect:view.do";
      }
      return setView ;
}
```

추천: BoardController 추가 (recommend.do)

```
@RequestMapping("/recommend.do")
public String updateRecommendCnt(HttpServletRequest request, HttpSession session, RedirectAttributes rea){

      int bno = Integer.parseInt(request.getParameter("bno"));
      String userId = (String) session.getAttribute("userId");

      if( userId==null){
            rea.addAttribute("bno", bno);
            rea.addAttribute("errCode", "4");
            return "redirect:/board/view.do";
      }

      BoardLikeVO boardLike = new BoardLikeVO();
      boardLike.setBno(bno);
      boardLike.setUserId(userId);

      BoardVO board = boardService.getArticle(bno);

      if( board.getWriterId().equals(userId)){
            rea.addAttribute("errCode", "3");                //본인 글은 추천 불가
      } else {
            if (boardService.getBoardLike(boardLike) == 0 ) {  //이미 추천한 기록이 없으면
                  boardService.increaseRecommendCnt(bno);
                  boardService.addBoardLike(boardLike);         //추천 기록 등록
            }else {
                  rea.addAttribute("errCode", "2");           //이미 추천했던 글이면 추천 불가
            }
      }
      rea.addAttribute("bno", bno);
      return "redirect:/board/view.do";
}
```

댓글쓰기: BoardController에 추가(replyWrite.do:)

```
@RequestMapping("/writeReply.do")
public String replyWriteProc(@ModelAttribute("reply") ReplyVO reply, RedirectAttributes rea){

    if (reply.getContent().isEmpty()) {
        rea.addAttribute("errCode", "1");
    }
    else {
        String  content = reply.getContent().replaceAll("<","&lt;");
        content = reply.getContent().replaceAll(">","&gt;");
        content = reply.getContent().replaceAll("&","&amp;");
        content = reply.getContent().replaceAll("\"","&quot;");
        content = reply.getContent().replaceAll("\r\n", "<br />");

        reply.setContent(content);

        boardService.writeReply(reply);
    }

    rea.addAttribute("bno", reply.getBno());

    return "redirect:view.do";
}
```

댓글 지우기

- BoardService ➔ BoardMapper.XML(::Mapper.Java)확인
- Controller 추가

```
@RequestMapping("/deleteReply.do")
public String commentDelete(HttpServletRequest request, HttpSession session,
RedirectAttributes rea){

    int rno = Integer.parseInt(request.getParameter("rno"));
    int bno = Integer.parseInt(request.getParameter("bno"));

    String userId = (String) session.getAttribute("userId");
    ReplyVO reply = boardService.getReply(rno);

    if(!userId.equals(reply.getWriterId())){
        rea.addAttribute("errCode", "1");
    } else {
        boardService.deleteReply(rno);
    }

    rea.addAttribute("bno", bno); // move back to the article

    return "redirect:view.do";
}
```

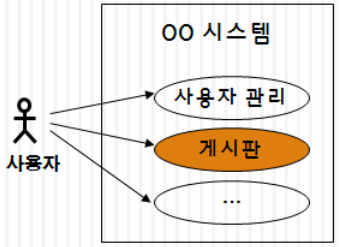
댓글 추천

```
@RequestMapping("/recommandReply.do")
public String RecommendRely(HttpServletRequest request, HttpSession session, RedirectAttributes rea){

      int bno = Integer.parseInt(request.getParameter("bno"));
      int rno = Integer.parseInt(request.getParameter("rno"));
      String userId = (String) session.getAttribute("userId");

      if( userId==null){
            rea.addAttribute("bno", bno);
            rea.addAttribute("errCode", "4");;
            return "redirect:/board/view.do";
      }

      ReplyLikeVO replyLike = new ReplyLikeVO();
      replyLike.setRno(rno);
      replyLike.setUserId(userId);

      BoardVO board = boardService.getArticle(bno);

      if(board.getWriterId().equals(userId)){
            rea.addAttribute("errCode", "3");
      } else {
            if (boardService.getReplyLike(replyLike) == 0 ) {
                  boardService.incReplyRecCnt(rno);
                  boardService.addReplyLike(replyLike);
            } else {
                  rea.addAttribute("errCode", "2");
            }
      }
      rea.addAttribute("bno", bno);
      return "redirect:/board/view.do";
}
```

파일 download

```
@RequestMapping(value = "filedown")
@ResponseBody
public byte[] downProcess(HttpServletResponse response,@RequestParam  String fileName )
throws IOException{

    File file = new File("c:/upload/" + fileName);
    byte[] bytes = FileCopyUtils.copyToByteArray(file); //SPRING 5.0 이상

    String fn = new String(file.getName().getBytes(), "iso_8859_1");

    response.setHeader("Content-Disposition", "attachment;filename=\"" + fn + "\"");
    response.setContentLength(bytes.length);

    return bytes;
}
```

클래스 다이어그램



Write
(write.jsp)

글저장 ()
재작성()
파일업로드()

List
(list.jsp)

글쓰기 ()
글보기()
검색()
페이지이동 ()

View
(view.jsp)

추천()
댓글쓰기()
댓글삭제()
수정()
삭제()
목록이동()

BoardController
(BoardController.java)

목록보기 ()
글쓰기 ()
글보기()
검색()
페이징()
글삭제()

글수정보기 ()
글수정저장()
추천증가()

댓글쓰기()
댓글삭제()

Modify
(modify.jsp)

글저장 ()
재작성()
파일업로드()

BoarService
(BoardService.java)

페이지목록보기 ()
페이징()
한글조회()

글쓰기 ()
글삭제()
글수정()

추천증가()
조회수증가()
댓글증가()

댓글리스트 조회()
댓글 한글조회
댓글쓰기()
댓글삭제()
댓글수정()

PageHelper
(Pagehelper.java)

페이징 문자열생성()

BoardMapper
(BoardMapper.java)

페이지목록보기 ()

한글조회()

글쓰기 ()
글수정()
글삭제()

추천수정()
조회수수정()
댓글수정()

댓글리스트 조회()
댓글 한글조회
댓글쓰기()
댓글삭제()
댓글수정()

Interceptor 수정

```
// login.do 이나 회원가입의 경우(/bbs/member/join.do) 이미 로그인 되어 있으면 로그인 필요없음
if(request.getRequestURI().equals("/bbs/login.do") || request.getRequestURI().equals("/bbs/ajaxlogin.do")){
    if(userId != null){
        response.sendRedirect(request.getContextPath() + "/board/list.do");
        return true; //list.do 실행
    } else {
        return true; //login.do 실행
    }
}

if(userId == null){
    response.sendRedirect(request.getContextPath() + "/ajaxlogin.do");
    return false; //현재 URL 페이지 실행 않고 login.do 실행
} else {
    return true; //현재 URL 페이지 실행
}
//
```

개인정보 보호법(고유식별정보의 처리 제한)

제24조(고유식별정보의 처리 제한) ① 개인정보처리자는 다음 각 호의 경우를 제외하고는 법령에 따라 개인을 고유하게 구별하기 위하여 부여된 식별정보로서 대통령령으로 정하는 정보(이하 "고유식별정보"라 한다)를 처리할 수 없다.

1. 정보주체에게 제15조제2항 각 호 또는 제17조제2항 각 호의 사항을 알리고 다른 개인정보의 처리에 대한 동의와 별도로 동의를 받은 경우
2. 법령에서 구체적으로 고유식별정보의 처리를 요구하거나 허용하는 경우

② 삭제 <2013. 8. 6.>

③ 개인정보처리자가 제1항 각 호에 따라 고유식별정보를 처리하는 경우에는 그 고유식별정보가 분실·도난·유출·위조·변조 또는 훼손되지 아니하도록 대통령령으로 정하는 바에 따라 암호화 등 안전성 확보에 필요한 조치를 하여야 한다. <개정 2015. 7. 24.>

④ 행정안전부장관은 처리하는 개인정보의 종류·규모, 종업원 수 및 매출액 규모 등을 고려하여 대통령령으로 정하는 기준에 해당하는 개인정보처리자가 제3항에 따라 안전성 확보에 필요한 조치를 하였는지에 관하여 대통령령으로 정하는 바에 따라 정기적으로 조사하여야 한다. <신설 2016. 3. 29., 2017. 7. 26.>

⑤ 행정안전부장관은 대통령령으로 정하는 전문기관으로 하여금 제4항에 따른 조사를 수행하게 할 수 있다. <신설 2016. 3. 29., 2017. 7. 26.>

패스워드 및 주민등록번호 암호화, 복호화

개인정보보호법

- **제24조의2(주민등록번호 처리의 제한)** ① 제24조제1항에도 불구하고 개인정보처리자는 다음 각 호의 어느 하나에 해당하는 경우를 제외하고는 주민등록번호를 처리할 수 없다. <개정 2016.3.29.>

  1. 법률에서 구체적으로 주민등록번호의 처리를 요구하거나 허용한 경우
  2. 정보주체 또는 제3자의 급박한 생명, 신체, 재산의 이익을 위하여 명백히 필요하다고 인정되는 경우
  3. 제1호 및 제2호에 준하여 주민등록번호 처리가 불가피한 경우로서 안전행정부령으로 정하는 경우

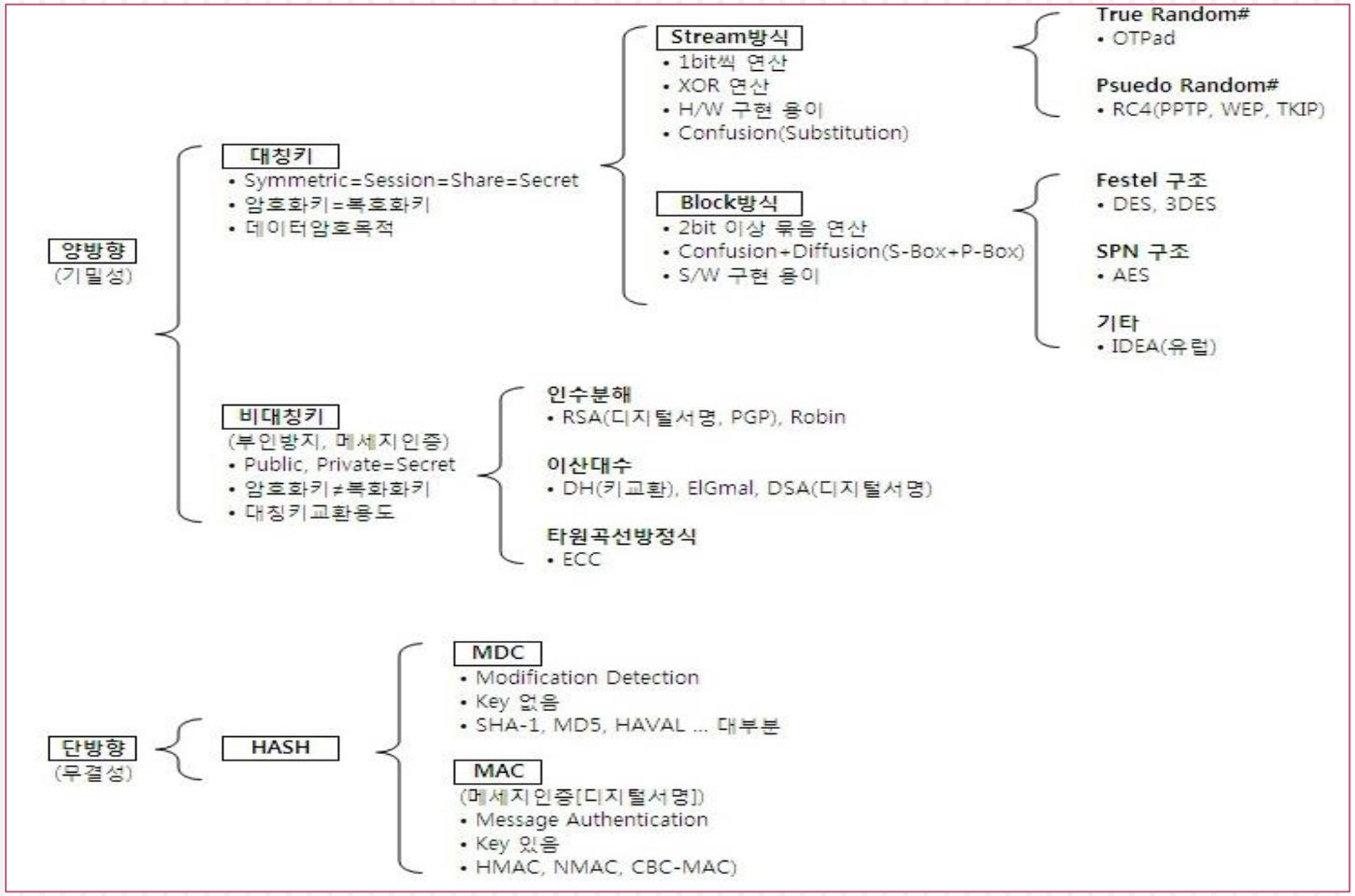
  ② 개인정보처리자는 제24조제3항에도 불구하고 주민등록번호가 분실·도난·유출·위조·변조 또는 훼손되지 아니하도록 암호화 조치를 통하여 안전하게 보관하여야 한다. 이 경우 암호화 적용 대상 및 대상별 적용 시기 등에 관하여 필요한 사항은 개인정보의 처리 규모와 유출 시 영향 등을 고려하여 대통령령으로 정한다. <신설 2014.3.24., 2015.7.24.>

  ③ 개인정보처리자는 제1항 각 호에 따라 주민등록번호를 처리하는 경우에도 정보주체가 인터넷 홈페이지를 통하여 회원으로 가입하는 단계에서는 주민등록번호를 사용하지 아니하고도 회원으로 가입할 수 있는 방법을 제공하여야 한다. <개정 2014.3.24.>

  ④ 안전행정부장관은 개인정보처리자가 제3항에 따른 방법을 제공할 수 있도록 관계 법령의 정비, 계획의 수립, 필요한 시설 및 시스템의 구축 등 제반 조치를 마련·지원할 수 있다. <개정 2014.3.24.>

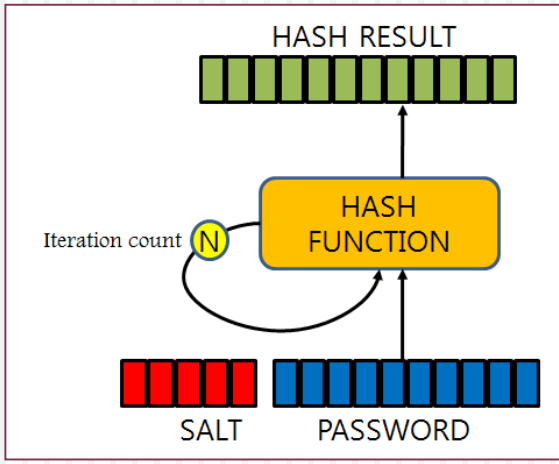
암호화 (Encryption) / Decryption

- 양방향 암호화 -_평문을 암호문으로 암호화 (Encryption) 하고 암호문을 다시 평문으로 복호화 (Decryption) 하는 암호화 기법 (주민등록번호)
- 단방향 암호화- 평문을 암호문으로 암호화하는 것은 가능하지만 암호문을 평문으로 복호화하는 것은 불가능한 암호화 기법. (패스워드)



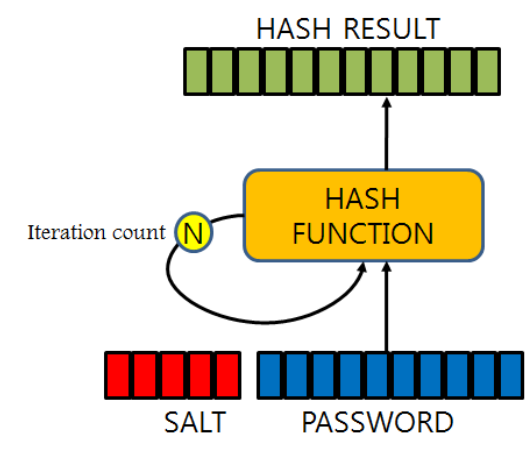
안전한 패스워드의 저장

- 단순 텍스트로 패스워드를 저장하는 것은 범죄를 저지르는 것이나 다름없다.
- 단방향 해시 함수
  - 수학적인 연산을 통해 원본 메시지를 변환하여 암호화된 메시지인 다이제스트를 생성한다
  - 원본 메시지를 알면 암호화된 메시지를 구하기는 쉽지만 암호화된 메시지로는 원본 메시지를 구할 수 없어야 하며 이를 '단방향성'이라고 한다.
- 단방향 해시 함수의 문제점
  - 동일한 메시지가 언제나 동일한 다이제스트를 갖는다면, 공격자가 전처리(pre-computing)된 다이제스트를 가능한 한 많이 확보한 다음 이를 탈취한 다이제스트와 비교해 원본 메시지를 찾아내거나 동일한 효과의 메시지를 찾을 수 있다. 이와 같은 다이제스트 목록을 레인보우 테이블(rainbow table)이라 하고, 이와 같은 공격 방식을 레인보우 공격(rainbow attack)이라 한다
  - 솔트(salt)는 단방향 해시 함수에서 다이제스트를 생성할 때 추가되는 바이트 단위의 임의의 문자열이다.
  - 동일한 패스워드를 사용하는 사용자의 다이제스트가 다르게 생성되어 인식 가능성 문제가 크게 개선된다.
  - 키 스트레칭(key stretching) -입력한 패스워드의 다이제스트를 생성하고, 생성된 다이제스트를 입력 값으로 하여 다이제스트를 생성하고, 또 이를 반복하는 방법으로 다이제스트를 생성할 수도 있다. 이렇게 하면 입력한 패스워드를 동일한 횟수만큼 해시해야만 입력한 패스워드의 일치 여부를 확인할 수 있다.



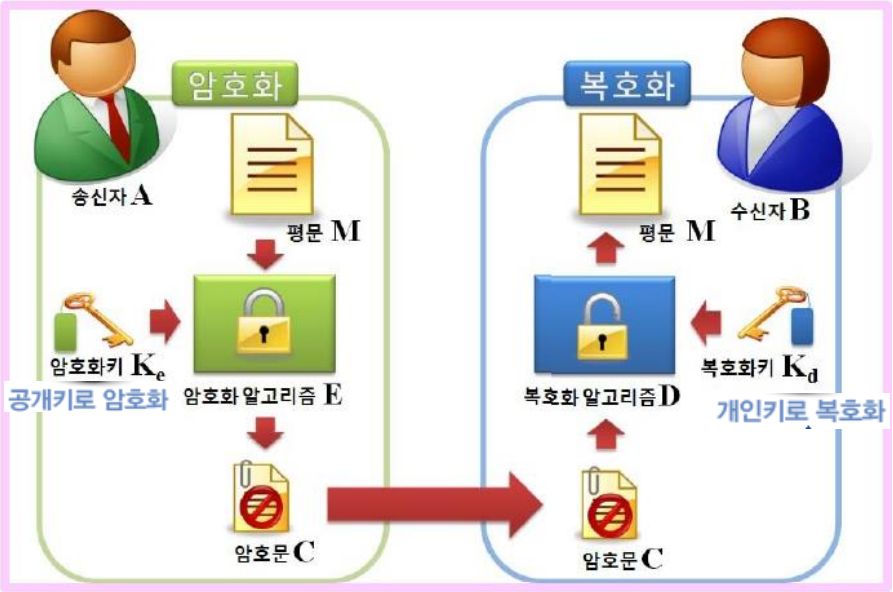
패스워드 암호화 알고리즘 종류

- Adaptive Key Derivation Functions
  - 다이제스트를 생성할 때 솔팅과 키 스트레칭을 반복하며 솔트와 패스워드 외에도 입력 값을 추가하여 공격자가 쉽게 다이제스트를 유추할 수 없도록 하고 보안의 강도를 선택할 수 있다.
- PBKDF2(Password-Based Key Derivation Function)
  - 해시 함수의 컨테이너인 PBKDF2는 솔트를 적용한 후 해시 함수의 반복 횟수를 임의로 선택할 수 있다. PBKDF2는 아주 가볍고 구현하기 쉬우며, SHA와 같이 검증된 해시 함수만을 사용한다.
- Bcrypt
  - bcrypt는 처음부터 패스워드 저장을 목적으로 설계되었다. Niels Provos와 David Mazi□res가 1999년 발표했고 현재까지 사용되는 가장 강력한 해시 메커니즘 중 하나이다. "work factor" 인자는 하나의 해시 다이제스트를 생성하는 데 얼마만큼의 처리 과정을 수행할지 결정한다. "work factor"를 조정하는 것만으로 간단하게 시스템의 보안성을 높일 수 있다.
- Scrypt
  - 다이제스트를 생성할 때 메모리 오버헤드를 갖도록 설계되어, 억지 기법 공격(brute-force attack)을 시도할 때 병렬화 처리가 매우 어렵다. 따라서 PBKDF2보다 안전하다고 평가되며 미래에 bcrypt에 비해 더 경쟁력이 있다고 여겨진다.



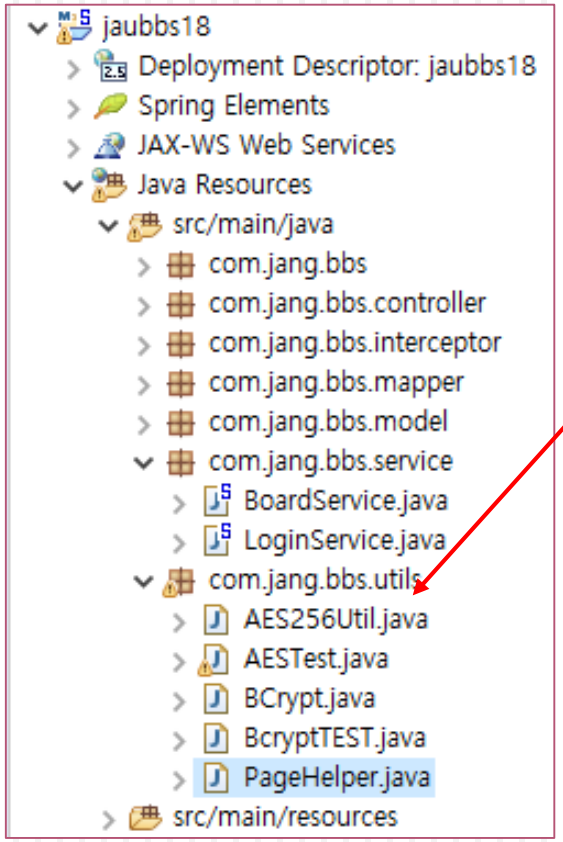
암호화 복호화(Encryption / Decryption)-양방향 해시함수

- 암호화 -허가된 자를 제외하고 제3자는 읽을 수 없도록 평문(Plaintext)을 부호화하여 암호문(Ciphertext)으로 만드는 것.
- 복호화 - 암호문을 평문으로 역부호화하는 것
- 암호화와 복호화를 위해 일반적으로 비밀 키를 이용한다. 암호화와 복호화 시 동일한 키를 이용하는 암호 시스템을 대칭키 암호시스템이라고 부르고, 서로 다른 두 개의 키를 이용하는 암호 시스템을 비대칭키 암호시스템이라고 부른다. 비대칭 키 암호 시스템의 예로 공개키 암호 시스템이 있다.
- **대칭키 암호 시스템의 종류.**
  - DES
  - Advanced Encryption Standard(AES)
  - ARIA
  - Twofish
  - SEED
- **공개키 암호 시스템의 종류**
  - RSA
  - ElGamal
  - ECC(Elliptic Curve Cryptosystem, -타원 곡선 암호 시스템)
  - 배낭 암호



[그림 2-1] 암・복호화 과정

암호화/복호화 구현

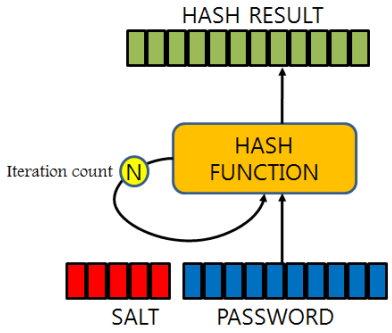


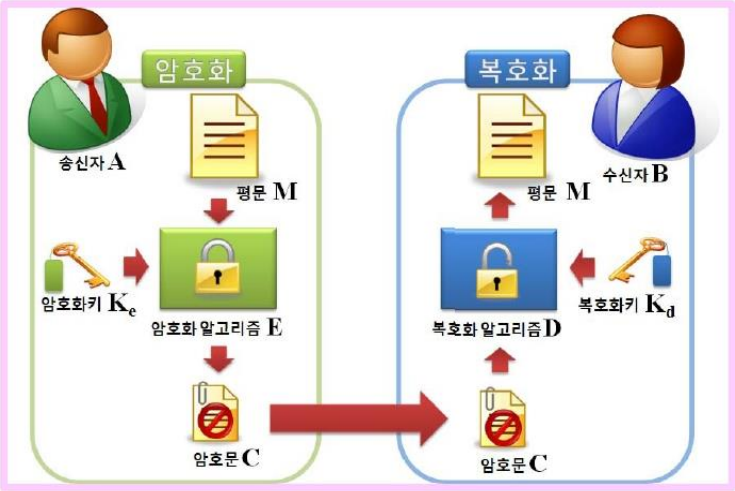
1. Bcrypt.java , AES256Util.java 파일 복사
2. 관련 test.java 복사
3. Commons-codec-1.11 pom 추가
4. 실행 테스트

```
<dependency>
    <groupId>commons-codec</groupId>
    <artifactId>commons-codec</artifactId>
    <version>1.11</version>
</dependency>
```

패스워드, 주민번호 암호화 저장

```
@RequestMapping(value="/join.do", method = RequestMethod.POST)//입력폼 내용 저장
public String onSubmit(@Valid UserVO userVO, BindingResult result,  Model model) throws Exception {

      if (result.hasErrors()) {
            model.addAllAttributes(result.getModel());
            return "/member/joinForm";
      }
      //passwd 암호화
      String hashPass = BCrypt.hashpw(userVO.getPasswd(), BCrypt.gensalt(12));
      userVO.setPasswd(hashPass);  //암호화된 passwd 로 변경

      //주민번호 암호화
      String jumin1=userVO.getJumin().substring(0,6);
      String jumin2=userVO.getJumin().substring(6);

      String key="jangan-1182-Key!!";
      AES256Util aes256 = new AES256Util(key);

      String hashjumin = aes256.aesEncode(jumin2);
      String encjumin=jumin1+"-"+hashjumin;

      userVO.setJumin(encjumin);

      if(this.loginService.insertUser(userVO) != 0)   {
            model.addAttribute("user", userVO);
            model.addAttribute("errCode", 3);// 등록성공
            return "/member/login";
      }
      else {
            model.addAttribute("errCode", 5);// 등록실패
            return "/member/joinForm";
      }
}
```





[그림 2-1] 암·복호화 과정

패스워드 체크

LoginService

```
public UserVO getUser(UserVO userVO) {
   return loginMapper.findUser(userVO);
}
```

```
public UserVO getUser(String userId) {
   return loginMapper.getUser(userId);
}
```

```
public UserVO findUser(UserVO userVO) {
          return loginMapper.findUser(userVO);
}
```

Mapper

```
UserVO getUser(String userId);

int updateUser(UserVO userVO);
int insertUser(UserVO userVO);
```

```
<select id="getUser" parameterType="String" resultType="UserVO">
      SELECT    *
       FROM     bbs_member
       WHERE    userid = #{userId}
</select>

<insert id="insertUser" keyProperty="mno">
    <selectKey keyProperty="mno" resultType="int" order="BEFORE">
       SELECT COALESCE(MAX(mno), 0) + 1 AS mno   FROM   bbs_member
    </selectKey>

    INSERT INTO bbs_member (mno, name, passwd, userid, jumin, email )
    VALUES ( #{mno}, #{name}, #{passwd}, #{userId}, #{jumin}, #{email})
</insert>

<update id="updateUser" parameterType="UserVO">
    UPDATE  bbs_member
    SET    passwd = #{passwd}, email = #{email}
    WHERE   mno  = #{mno}
</update>
```

패스워드 체크(controller)

```
@RequestMapping(value="/login.do", method = RequestMethod.POST)
public String loginProc(@Valid UserVO userVO, BindingResult result, Model model, HttpSession session) {

      if (result.hasFieldErrors("userId")||result.hasFieldErrors("passwd"))  {
            model.addAllAttributes(result.getModel());
            return "/member/login";
      }
      UserVO loginUser = this.loginService.getUser(userVO.getUserId());

      if(loginUser == null){
            model.addAttribute("userId", "");
            model.addAttribute("errCode", 1);// 등록되지않은 아이디

            return "/member/login";
      }
      else if(BCrypt.checkpw(userVO.getPasswd(), loginUser.getPasswd()) ) { //패스워드 일치
            model.addAttribute("loginUser", loginUser);

            session.setAttribute("userId", loginUser.getUserId()); //세션에 변수등록
            session.setAttribute("userName", loginUser.getName()); //세션에 변수등록
            return "redirect:/board/list.do";
      }
      else {
            model.addAttribute("userId", "");
            model.addAttribute("errCode", 4);// 패스워드 불일치

            return "/member/login";
      }

}
```

정보수정(주민등록번호 복호화)

```
@RequestMapping(value="/edituser", method = RequestMethod.GET)
public String toUserEditView( HttpSession session, Model model) throws
    java.io.UnsupportedEncodingException,
    NoSuchAlgorithmException,
    NoSuchPaddingException,
    InvalidKeyException,
    InvalidAlgorithmParameterException,
    IllegalBlockSizeException,
    BadPaddingException  {

      String userId = session.getAttribute("userId").toString();
      UserVO loginUser = this.loginService.getUser(userId);

      if(loginUser == null){
            model.addAttribute("userId", "");
            model.addAttribute("errCode", 1);// 등록되지않은 아이디

            return "/member/login";
      }
      else {//주민번호 복호화
            String jumin1=loginUser.getJumin().substring(0,6);
            String jumin2=loginUser.getJumin().substring(7);

            String key="jangan-1182-Key!!";
            AES256Util aes256 = new AES256Util(key);

            String decjumin = aes256.aesDecode(jumin2);
            decjumin = jumin1 + "-" + decjumin;

            loginUser.setJumin(decjumin);
             model.addAttribute("userVO", loginUser);
             return "/member/editForm";
      }
}
```

정보수정(패스워드 암호화 저장)

```
@RequestMapping(value="/edituser", method = RequestMethod.POST)
public String onEditSave( @Valid UserVO userVO, BindingResult result, Model model)
throws Exception {

      if (result.hasErrors()) {
          model.addAllAttributes(result.getModel());
          return "/member/editForm";
      }

      userVO.setPasswd(BCrypt.hashpw(userVO.getPasswd(), BCrypt.gensalt(12)));

      try {
          this.loginService.updateUser(userVO);
          model.addAttribute("userVO", userVO);
          return "/board/list.do";
      }
      catch ( DataAccessException e) {
          result.reject("error.duplicate.user");
          model.addAllAttributes(result.getModel());
          return "/member/editForm";
      }
}
```

View 수정

list.jsp

```
<ul>
    <li><a href="list.do">스프링 게시판</a></li>
    <li><a href="../member/logout.do">로그아웃</a></li>
    <c:if test="${not empty userId}">
      <li><a href="../member/edituser.do">정보수정</a></li>
    </c:if>
</ul>
```

joinForm.jsp 생성

패스워드 찾기

아이디 찾기

실행 및 DB 확인